

PULSE UNA LETRA QUE DESCRIBA EL PROBLEMA:
A - EL MOTOR NO GIRA
B - EL MOTOR GIRA PERO NO ARRANCA
C - ARRANCA, ENTONCES SE CALA
) OK
(S - SI, N - NO)?
) OK S
VIFI ANAYA
Tim Hartnell
Sistemas Expertos
Introducción al diseño y aplicaciones
Listados para MSX, Spectrum, Amstrad Commodore 64, Apple II
Incluye Intérpretes de LISP y PROLOG en Basic

SISTEMAS EXPERTOS

Introducción al diseño y aplicaciones

Sistemas expertos

Introducción al diseño y aplicaciones

Tim Hartnell



MICROINFORMATICA

Título de la obra original:
EXPLORING EXPERT SYSTEMS ON YOUR MICROCOMPUTER

Traducción: Eduardo Michelena y Francisco Serón
Diseño de colección: Antonio Lax
Diseño de cubierta: Narcís Fernández
Revisión de programas: Gustavo Flores

Villafranca, 22. 28028 Madrid
Depósito legal: M. 17.302-1986
ISBN: 84-7614-091-6
Printed in Spain
Imprime: Anzos, S. A. - Fuenlabrada (Madrid)

Indice

Apéndices

Los listados incluidos en el cuerpo principal del texto se han diseñado para los ordenadores MSX.

En los apéndices del libro encontrará listados específicos para las siguientes máquinas:

- **Spectrum.**
- **Amstrad.**
- **Commodore 64.**
- **Apple II.**

Introducción

Bienvenido a este trabajo sobre el fascinante mundo de los sistemas expertos. En él hablaremos sobre los jalones específicos en la historia y el desarrollo de algunos de los sistemas principales desde un punto de vista que nos permitirá obtener una visión de conjunto de la situación actual de los sistemas expertos.

A partir de ello, desarrollaremos sistemas expertos específicos propios (incluyendo MECANICO DE AUTOMOVILES, y MEDICI, que le harán un rápido "test" del estado de su maquinaria y una predicción de la edad probable hasta la que vivirá). El sistema experto más importante del libro se llama FUZZY RITA. RITA nos aporta la estructura de un sistema experto de propósito general que podemos modificar para adaptarlo a nuestras necesidades concretas. Veremos a RITA en acción, distinguiendo entre perros y gatos, utilizando las propiedades de un metal para decirnos qué es y, finalmente, prediciendo el tiempo, en base a datos reales (y consiguiendo un índice de aciertos muy superior al que probablemente se obtendría haciéndolo solamente al azar).

Veremos también los lenguajes que están comenzando a dominar el mundo de los sistemas expertos y el de la inteligencia artificial. He escrito para usted emuladores en BASIC del LISP y del PROLOG que puede cargarlos en su ordenador para hacerse una idea de cómo se trabaja con tales lenguajes. Antes le proporcionaré un lenguaje nuevo y sencillo, de nombre HASTE, para que le introduzca brevemente en los ambientes de programación habitados por el LISP y el PROLOG.

En conjunto, este libro se ha diseñado para ayudarle a conseguir una base razonable de conocimientos propia, de manera que pueda ejercitar su propio saber sobre sistemas expertos.

TIM HARTNELL



1
¿Qué es un sistema experto?

Un sistema experto es un programa de ordenador que contiene saber humano sobre un tema, mantenido de tal forma que los no expertos pueden acceder a él y utilizarlo. Tal sistema se compone de dos partes principales: la información almacenada (el saber o pericia) y el procedimiento de razonamiento (que formula preguntas al usuario y toma decisiones en base a la información que ha recibido).

Los sistemas expertos pueden hacer muchas cosas. Pueden diagnosticar enfermedades infecciosas (MYCIN), deducir estructuras moleculares a partir de espectrogramas de masas (DENDRAL), buscar petróleo y metales preciosos (PROSPECTOR) y ayudarle a poner en marcha su coche por la mañana (MECANICO DE AUTOMOVILES, que veremos después en este libro).

Las características que diferencian un sistema experto-inteligente de una base de conocimientos son las siguientes (estos términos le resultarán comprensibles cuando se haya adentrado en este libro):

1. *Dominio específico.*
2. *Utiliza un razonamiento probabilístico o difuso.*
3. *Posee un modo autoexplicativo.*
4. *Separa nítidamente los hechos del mecanismo de inferencia.*
5. *Puede crecer incrementalmente.*
6. *Da dinero.*

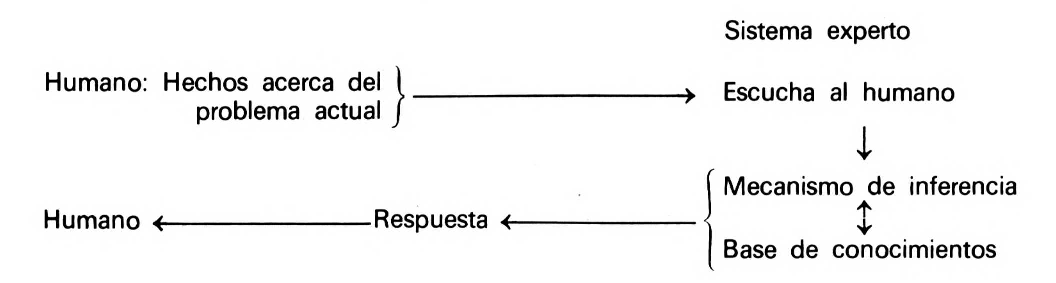
7. *Normalmente se basa en reglas.*
8. *No tiende a ser ambicioso.*
9. *Da como respuesta un consejo (tal como "cambia las bujías", "opera el dedo gordo del pie izquierdo", "cava en la esquina nordeste del huerto"), en vez de cualquier otro tipo de salidas (como, por ejemplo, un gráfico).*

A decir verdad, no sabemos todavía qué es un sistema experto. Hoy día es demasiado pronto para dar una respuesta definitiva a una pregunta aparentemente tan simple: ¿qué es un sistema experto? Somos incapaces de decir cuáles de las características que acabamos de relacionar se convertirán en fundamentales y cuáles acabarán siendo accesorias.

Hasta ahora, el mayor problema ha sido conseguir el conocimiento de un experto de forma que una máquina pueda manipularlo. El procedimiento actual, llamado "ingeniería del conocimiento", está lejos de ser eficiente, ya que requiere un experto humano (el ingeniero de conocimiento) para extraer laboriosamente el conocimiento real que otro experto humano concentra en un problema. Este es el "cuello de botella" en la producción de sistemas expertos.

Los componentes principales

A modo de diagrama, las partes principales de sistema experto son las siguientes:



La "base de conocimientos", en un sistema basado en regla (tal como MECANICO DE AUTOMOVILES de este libro), está formada básicamente por una colección de sentencias IF/THEN (SI/ENTONCES), tales como:

SI X E Y SON VERDADEROS ENTONCES C ES VERDADERO

MECANICO DE AUTOMOVILES contiene líneas de códigos que significan SI EL ARRANQUE NO HACE UN CHASQUIDO ENTONCES EL RELE PROBABLEMENTE FALLA. Funciona pidiendo primeramente al usuario que defina el problema. A continuación lo conduce descendiendo por el

árbol de posibilidades, tomando una u otra rama en función de las informaciones que suministra el usuario. Este es un ejemplo de "encadenamiento hacia adelante" que busca la solución única que se encuentra al extremo de una serie particular de bifurcaciones.

Otros sistemas expertos más sofisticados utilizan un "encadenamiento hacia atrás" en los que, elegido un fin (EL COCHE NO TIENE GASOLINA), se formulan las preguntas para establecer la veracidad de esa hipótesis particular.

La base de conocimientos

Este puede hallarse o codificado *(hard-wired)** directamente en el cuerpo principal del programa (como en MEDICI y MECANICO DE AUTOMOVILES), en cuyo caso es más o menos imposible borrarlo o mantenerlo en una base de datos direccionable, de forma que pueda ser accedido, modificado y actualizado, incluso mientras el programa está en funcionamiento. FUZZY RITA establece un conjunto de reglas que determinan la importancia que se les dará a determinadas entradas, asignando valores a variables (y modificándolos en base a la aplicación en sucesivas veces del método de la prueba y corrección del error).

En el caso de un sistema *hard-wired* (y en total contraste con los programas del tipo de FUZZY RITA que generan sus propias reglas mientras están funcionando, incluso si el usuario no tiene ni idea de cómo son éstas), el saber debe adquirirse directamente de un experto humano.

La puesta a punto de un método mediante el cual un ordenador pudiera extraer por sí mismo el conocimiento, y así crear su propia base de conocimiento, sería tal vez el avance más importante en la historia de los sistemas expertos. Un conocimiento autocreado de este tipo sería rápidamente reconocido como un valioso recurso. Los que consigan satisfacer la demanda de tal producto obtendrán una fortuna. (La captación del saber de expertos humanos se discute con algún detalle en el capítulo 7, "Knowledge Engineering", del libro *Rule-Based Expert System*, B. Buchanan, 1984.)

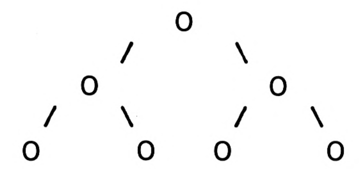
A lo ancho del mundo se están desarrollando varios proyectos de investigación sobre el tema de "el aprendizaje de máquinas", que es la ciencia cuyo objetivo es la consecución de un ordenador que descubra y codifique para él mismo el saber humano. (Para informarse de cómo trabaja un ingeniero del conocimiento, léase H. Penny Nii, Feigenbaum, 1983.)

A continuación damos algunas de las estructuras del razonamiento utilizadas en los sistemas expertos actuales.

1. ARBOLES DE DECISION. El ordenador sigue una trayectoria rígida descendente según una sucesión de preguntas, dejando que la respuesta a cada una de ellas determine la trayectoria específica que el programa deberá seguir a continuación. Esta clase de programas necesitan datos

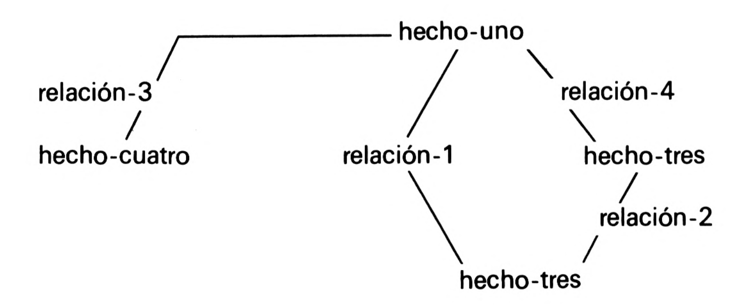
* *Hard-wired:* cableado, de estructura rígida, etc.

rígidos, les gustan situaciones SI/NO. No son capaces de arreglárselas con facilidad con situaciones difusas del tipo "habitualmente/quizá/a veces".



Programas comerciales tales como CONSULT y EXPERTEASE utilizan árboles de decisión.

2. PRODUCCION DE REGLAS. Esta estructura se ve en programas comerciales como XCON y DART, que utilizan procesos IF/THEN (junto con AND).
3. REDES SEMANTICAS. MYCIN y PROSPECTOR son ejemplos de este tipo de estructura:



Los sistemas expertos y la inteligencia artificial (IA) son dos temas de gran actualidad. En la edición de enero de 1985 de la revista *Byte* (págs. 275 a 282), bajo el título de *Expert Systems — Myth or Reality? (Sistemas expertos: ¿Mito o realidad?)*, Bruce D'Ambrosio señala que el Departamento de Defensa de Estados Unidos ha clasificado la IA como una de las diez tecnologías más importantes que deberían ser desarrolladas en los próximos años de este siglo. Japan Inc. ha entrado en acción con su "proyecto de ordenador de la quinta generación". Uno de los principales objetivos de este proyecto es el desarrollo de mejores sistemas expertos.

Entre los temas principales para la investigación y desarrollo de los ordenadores de la quinta generación (de acuerdo con los *Proceedings of the International Conference on Fifth-Generation Computer Systems*, 19-22 de octubre de 1981, Japan Information Processing and Development Centre) se hallan incluidos los siguientes:

SISTEMAS DE APLICACION BASICA	Traducción automática Preguntas-respuestas Aplicación del reconocimiento de voz Reconocimiento de dibujos e imágenes Resolución de problemas
SISTEMAS DE *SOFTWARE* BASICO	Control de la base de conocimientos Resolución de problemas e inferencia Interfaz inteligente
NUEVAS ARQUITECTURAS AVANZADAS	Máquinas de programación lógica Máquinas funcionales Máquinas con álgebra relacional Máquinas con soporte de datos de tipo abstracto Máquinas con flujo de datos
ARQUITECTURA EN RED CON FUNCIONES DISTRIBUIDAS	Máquina base de datos Máquina con alta velocidad de computación numérica Sistema de comunicación hombre-máquina de alto nivel
TECNOLOGIA VLSI	Arquitectura VLSI (muy alta escala de integración) Arquitectura inteligente VLSI CAD (diseño asistido por ordenador)
TECNOLOGIA DE SISTEMIZACION	Sistemas de programación inteligente Sistemas de diseño de bases de conocimientos

Espoleadas por la tentativa de los japoneses, Francia y Gran Bretaña tienen ahora proyectos nacionales de investigación. (¡Tal vez pueda usted contribuir a la investigación en su propio país con las experiencias que obtenga de este libro!) En el próximo capítulo veremos cómo se estilizan en la práctica algunas de las ideas que hemos bosquejado.



2 Sistemas expertos en funcionamiento

Los tres sistemas expertos que vienen a continuación son actualmente los más conocidos y experimentados:

- MYCIN, utilizado en el campo de las enfermedades infecciosas.
- DENDRAL, que analiza componentes de química orgánica.
- PROSPECTOR, que sugiere el emplazamiento de yacimientos minerales.

En este capítulo veremos los tres y aprenderemos de ellos algunas características generales. Dedicaremos más atención a MYCIN, ya que parece ser el más importante.

Los experimentos MYCIN en Stanford

Comenzaremos con el sistema experto de Stanford, una poderosa criatura que puede resultar útil en el diagnóstico de enfermedades infecciosas. MYCIN es uno de los sistemas expertos más utilizados actualmente, y de su estudio podemos extraer muchas conclusiones importantes. Partiendo de la información de esta sección, probablemente descubra maneras de desarrollar y ampliar su propio sistema experto.

MYCIN es uno de los sistemas expertos más antiguos que continúan funcionando. Trabaja apoyándose en un método basado en reglas, en áreas en las que

anteriormente no se creía que los datos involucrados en la toma de decisiones pudieran ser codificables.

MYCIN fue precedido en el Stanford Heuristic Programming Project (Proyecto Stanford de Programación Heurística) por DENDRAL, que es capaz de deducir las estructuras moleculares en base a los datos suministrados por un espectrógrafo de masas. No obstante, DENDRAL no se diseñó desde un principio para ser un sistema experto, sino que evolucionó hasta llegar a serlo como resultado de una investigación sobre inducción científica. Además, DENDRAL trabaja en un campo en el que los procesos involucrados para la toma de una decisión están claramente establecidos antes de que el sistema se construyera. MYCIN se inició como un proyecto para la construcción de un sistema experto, en un campo donde las reglas para la toma de decisiones eran difusas y mal definidas, y el proyecto triunfó. Volveremos a ver a DENDRAL con cierto detalle a su debido tiempo.

MYCIN no está diseñado para sustituir a un médico, sino para desempeñar el papel de consejero. El paciente es examinado por el médico que comunica a MYCIN los síntomas observados. MYCIN entonces da su diagnóstico, que es muy parecido al que podría dar un especialista en las mismas circunstancias. Además de ello, MYCIN puede explicar el camino que ha seguido para alcanzar su conclusión. Un desarrollo de este aspecto del comportamiento de MYCIN es un sistema experto que instruye a estudiantes de medicina, creando así nuevos expertos a partir del saber codificado de otros que les preceden. (Una importante fuente de información sobre el proyecto MYCIN es *Rule-Based Expert Systems*, Buchanan, 1984.) Si desea estudiar en detalle la evolución del proyecto y las ideas que están detrás, debería leer este libro.

Un desarrollo posterior de MYCIN, llamado EMYCIN (de *Essential MYCIN)*, se creó suprimiendo el saber codificado para la formulación de diagnósticos (la base de conocimientos) y dejando el motor de inferencia (el bloque deductivo). Esto ha permitido crear, con más o menos simplicidad, sistemas expertos utilizables en otros temas, ya que se puede acoplar a EMYCIN nuevas bases de conocimientos. Esto nos sugiere que hay cosas que se pueden aprender de la estructura de EMYCIN y que pueden aplicarse de forma general a los sistemas expertos basados en reglas.

El equipo MYCIN

Los creadores de MYCIN (que con el tiempo han venido a autodenominarse "el equipo MYCIN") comenzaron su trabajo sobre el sistema con el propósito de seguir la iniciativa de un programa llamado SCHOLAR, que estaba capacitado para "hablar" muy bien sobre la geografía de América del Sur. No obstante, se encontraron con que los datos médicos que utilizaban no encajaban en una estructura ordenada de forma tan obvia como lo hacían los datos geográficos, por lo que la línea SCHOLAR se abandonó.

La experimentación demostró que los desarrollos más fructíferos apuntaban en la dirección de la codificación explícita de reglas específicas —tal como las

utilizan los expertos humanos— que unen elementos de una base de datos, en vez de intentar desplazar tales reglas para crear trayectorias generales que pudieran seguir el proceso de toma de decisiones. El mantener las reglas de esta manera implica también que MYCIN puede explicar cómo ha llegado a una conclusión dada, utilizando el tipo de frases que los mismos médicos utilizarían. (Entre las personas relacionadas con el desarrollo de MYCIN se encuentran J. Aikins, S. Axline, J. Bennett, A. Bonnet, B. Buchanan, W. Clancey, S. Cohen, R. Davis, L. Fagan, F. Rhame, C. Scott, E. Shortliffe, W. van Melle, S. Wraith y V. Yu).

Las reglas

El primer problema grande que tuvo que acometer el equipo, una vez que la naturaleza de la regla de base había evolucionado, fue que el conocimiento humano de las enfermedades infecciosas no encajaba en paquetes ordenados mutuamente exclusivos. Frecuentemente la barrera entre un elemento y el siguiente dentro de un campo estaba definida de forma difusa.

La primera implementación del programa disponía de 200 reglas, que funcionaban en esencia como si fueran líneas mantenidas en la forma de una sentencia IF seguida de un THEN (y en ocasiones incluyendo un ELSE).

En sesiones posteriores se suprimieron los ELSE, ya que se comprobó que se utilizaban en ocasiones muy raras. Actualmente MYCIN dispone de más de 600 reglas.

Se decidió que MYCIN debería tomar decisiones según el orden siguiente:

- Decidir qué organismo (si es que había alguno) era el causante significativo de la infección.
- Encontrar sobre qué órgano actuaba.
- Decidir qué medicamentos podrían utilizarse.
- Sugerir el mejor tratamiento.

A continuación damos una regla MYCIN típica:

Si: 1) el tinte del organismo es gram-positivo;
2) la morfología del organismo es coco, y
3) la formación de crecimiento del organismo es agrupada,
entonces hay una evidencia (0.7) de que la identidad del organismo es esfilococo.

Observe que el número que sigue a “evidencia” (0.7) indica el grado de certeza que MYCIN atribuye a su conclusión.

Este grado de certeza se expresa en una escala que va de -1 a $+1$, en la que -1 es el caso “falso” (desaprobación absoluta de la hipótesis), se pasa por 0 (no hay evidencia ni a favor ni en contra de la hipótesis, o la evidencia señala por igual ambas posibilidades) y se llega hasta 1 (la hipótesis ha sido probada).

Los autores dicen que éste es un sistema imperfecto, pero ha funcionado apreciablemente bien en el pasado y advierten que debería considerarse únicamente como un punto de partida en el camino que lleve a la construcción de un sistema lógico y consistente para la toma de decisiones, en áreas donde los datos son inciertos, y el dominio es complejo (Shortliffe, 1983).

La regla se presenta al médico de una manera fácil de comprender tal y como se ha expuesto anteriormente. No obstante, internamente las reglas se mantienen como listados de LISP, de la manera siguiente:

```
PREMISE: ($AND                 (SAME CNTXT GRAM GRAMPOS)
                               (SAME CNTXT MORPH COCCUS)
                               (SAME CNTXT CONFORM CLUMPS)
ACTION: (CONCLUDE CNTXT IDENT STAPHYLOCOCCUS TALLY .7)
```

(Probablemente le resultará más fácil entender el listado anterior una vez haya leído los apartados sobre EASLE y LISP-A de este libro.)

A menudo, un sistema experto será utilizado por personas que son menos expertas que aquellas cuyo saber está codificado en el sistema. Esto significa que el razonamiento que conduce a las conclusiones alcanzadas no siempre será evidente. MYCIN dispone de un subprograma denominado CHRONICLER que puede responder a preguntas relativas a la terapia recomendada. Un diálogo con CHRONICLER tendría el siguiente aspecto:

** **¿POR QUE RECETO EL FARMACO "X" PARA LA SITUACION "Y"?**
"X" se receta para "Y"
Porque
X es a menudo usado para Y en la enfermedad Z.

- **El elemento Q de los datos del paciente muestra intolerancia al fármaco P.**
- **X no está contraindicado.**

La ingeniería del conocimiento es un proeso lento y caro. El programa TEIRESIAS fue desarrollado para permitirle a MYCIN mejorar su base de datos ampliando conocimientos y eliminando errores, inconsistencias y omisiones. Cuando un médico descubre un diagnóstico erróneo o incompleto, puede poner en funcionamiento a TEIRESIAS para que formule una serie de preguntas con las que se creará una nueva regla MYCIN.

MYCIN se ha diseñado para comunicarse con el médico que la utiliza, de una manera tan natural como sea posible, y esto se ve claramente en una conversación con TEIRESIAS. Las salidas que da el ordenador durante el proceso de formación de reglas incluye frases tan "humanas" como las siguientes:

Me fastidia corregirle, doctor..., pero sabía que la mayor parte de las reglas en esta área incluyen...

Intentaré escribir una cláusula que abarque...

Una vez que las reglas están en su sitio, y han sido comprobadas y depuradas, MYCIN las compila dentro de una estructura arbórea, y seguidamente la compila en código máquina.

EMYCIN

MYCIN tiene dos componentes principales: la base de conocimiento y el bloque de inferencia. En 1974, dos años después de que el proyecto MYCIN comenzara, los primeros experimentos para la sustitución de la base de conocimiento para diagnósticos por otras relativas a campos distintos estaban subyacentes. Para proporcionar al bloque de inferencia material con el que trabajar se utilizó como fuente de saber un *Pontiac Service Manual* y se dieron quince reglas relativas al circuito del claxon. (Aunque su estructura es bastante diferente, la naturaleza del conocimiento encerrado en el programa MECANICO DE AUTOMOVILES de este libro se asemeja mucho a las reglas dadas para el primer "hijo de MYCIN".)

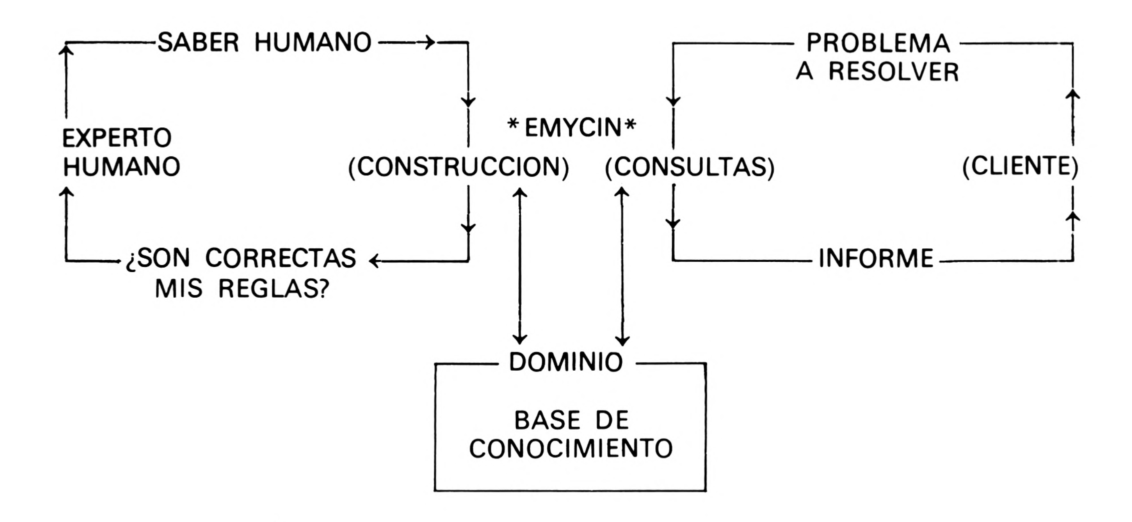
La experiencia adquirida en el campo *Pontiac*, y en posteriores experimentos, condujo al desarrollo de EMYCIN *(Essential MYCIN)*. Se trata de un bloque de inferencia de estructura MYCIN, junto con una parte importante del sistema TEIRESIAS (que, como recordará, se utiliza para añadir nuevas reglas a la base de conocimientos mediante una "conversación" directa con un usuario no-programador).

Si ejecutamos EMYCIN con la intención de crear nuestro propio sistema experto, pasaremos a través del proceso de creación paso a paso, en un diálogo que a veces es como el siguiente:

— ¿Desea crear una nueva base de conocimientos?
— Introduzca una palabra o frase que describa su dominio:
— Introduzca un nombre de una palabra para la raíz de árbol, el objeto central con el que las consultas se van a realizar:

Una vez que la información está en su sitio, EMYCIN se convierte en un experto en ese ámbito, y puede utilizarse como un sistema experto. Así pues, el programa tiene dos cometidos completamente diferentes. El primero es el de admitir datos y elaborar reglas a partir de ellos, y el segundo, el de tomar decisiones en base a tales reglas.

Seguidamente expresamos los cometidos de EMYCIN en forma de diagrama:



Uno de los mayores problemas a la hora de crear sistemas expertos nuevos partiendo de EMYCIN es la carencia de precisión en muchos de los enunciados de cualquier idioma. Por supuesto, tal carencia puede cuantificarse, hasta cierto punto, al hacer las reglas (como vimos en MYCIN), pero no siempre es posible, especialmente cuando EMYCIN pueda estar interpretando incorrectamente los datos en bruto de la regla introducida. James Bennett (Buchanan, 1984) está desarrollando actualmente un sistema experto (denominado ROGET) que toma parte en la creación de nuevos sistemas expertos con EMYCIN, lo que debería llevar a una producción más efectiva de hijos de MYCIN.

A partir de EMYCIN se han desarrollado ya algunos sistemas expertos muy efectivos. Entre ellos están SACON, CLOT y PUFF (Buchanan, 1984). SACON pretende descubrir un *análisis estratégico* para un problema particular que incluye una estructura específica, mientras que CLOT proporciona informes médicos sobre problemas sanguíneos. Otro programa de medicina, PUFF, genera informes sobre enfermedades pulmonares, mientras que dos descendientes de PUFF, CENTAUR y WHEEZE, utilizan la base de conocimientos de PUFF, pero manipulándola con estructuras de representación y control diferentes de las suyas. Otro programa, VM (de *Ventilation Manager:* gestor de oxigenación), está al tanto del estado posoperativo de un paciente al que se le ha practicado una intervención cardiovascular de cirugía mayor, cuando la respiración asistida se necesita con una frecuencia elevada.

DENDRAL, un químico prometedor

DENDRAL, que se escribió antes que MYCIN, es un programa que analiza los compuestos orgánicos para determinar su estructura. Este sistema experto, desarrollado por E. A. Feigenbaum y J. Lederberg, en Standord, es tan efectivo en su cometido que se cree que ningún químico humano podría hacerlo

tan bien. Algunos de los análisis de DENDRAL han sido incluso publicados como resultados originales de investigación (Rich, 1983). Ha detectado también errores en tablas químicas publicadas (Raphaele, 1976).

El programa intenta deducir la composición química y la estructura orgánica de compuestos químicos, utilizando los datos suministrados por un espectrograma de masas de la sustancia y las lecturas de resonancia magnética nuclear.

DENDRAL sigue básicamente un proceso de tres pasos para la determinación de la estructura más probable del componente que está analizando. Estos tres pasos son proyecto, generación y test (Buchanan, 1981).

En la etapa de planificación, el sistema utiliza sus reglas internas para limitar el tipo de estructuras que serán comprobadas, eliminando las que son incompatibles con los datos dados. En la segunda etapa, se producen las posibles respuestas, generadas conforme a las reglas básicas de DENDRAL. La etapa final consiste en la comprobación de esta estructura. Si no fuera por la primera etapa, se podrían generar millones de estructuras posibles, lo que llevaría al sistema a necesitar un tiempo exagerado para dar una respuesta. En esta etapa se buscan las posibilidades con mucha meticulosidad, para asegurar que existen razones químicas de peso que justifiquen cualquier eliminación.

DENDRAL puede localizar los elementos más importantes de los datos comprobados y utilizarlos en conjunción con su conocimiento básico sobre la formación de estructuras químicas. Comprueba sus hallazgos mediante un "espectrómetro de masas simulado" que hace comprobaciones para ver si las respuestas potenciales podrían producir el mismo espectrograma que la sustancia que está siendo comprobada. Lo mismo que MYCIN, DENDRAL utiliza un grupo de reglas empíricas que le han sido dadas por expertos humanos.

Las lecturas del espectrómetro de masas constituyen una de las más importantes unidades de datos con las que DENDRAL trabaja. Un espectrómetro separa átomos y moléculas de acuerdo a sus masas diferentes, cargándolas eléctricamente de forma que campos magnéticos y/o eléctricos se pueden utilizar para separarlos. El espectrómetro produce un gráfico en el que los picos están relacionados con las poblaciones de iones particulares en la muestra. (Una introducción sencilla sobre el funcionamiento de un espectrómetro se puede encontrar en Whitfield, R. C., *Spectroscopy in Chemistry*, Longmans, Londres y Harlow, Green & Co., 1969, págs. 61-71.)

DENDRAL analiza el espectrograma de una sustancia a identificar, y utiliza sus reglas condición-acción para producir una lista de subestructuras que deben aparecer en la sustancia en cuestión y otra lista de las que no deben hacerlo.

A continuación damos una regla de DENDRAL (número 75; Winston, 1984). Como puede ver, es de la misma "familia" que las reglas de MYCIN que examinamos anteriormente:

Si hay un pico alto en 71 unidades de masa atómica
hay un pico alto en 43 unidades de masa atómica
hay un pico alto en 86 unidades de masa atómica
hay cualquier pico en 58 unidades de masa atómica
entonces debe haber una subestructura N-PROPYL-KETONE3

El mecanismo generador de DENDRAL se denomina CONGEN, y toma como datos de entrada las lecturas del espectrómetro de masas y de otros tests químicos. Además, el generador permite imponer límites en las estructuras que producirá, tales como el número de átomos de cada tipo en la molécula y cualquier otra limitación determinada por el operador (Buchanan, 1981). A partir de ellos, CONGEN elabora una lista completa de todas las estructuras posibles en base a los datos introducidos.

Aunque DENDRAL no puede utilizarse en toda su capacidad por aquellos que no estén altamente entrenados en las áreas particulares de la química que constituyen su dominio, y podría decirse, por tanto, que no satisface uno de los requisitos de un sistema experto (poner el saber el alcance de los no expertos), opera en este campo mejor que el mejor de los expertos humanos, aunque sabe mucho menos que ellos (Simons, 1984). Este sistema ha obtenido valiosos resultados en una variedad muy amplia de casos, incluyendo (Buchanan, 1981) la determinación de estructuras químicas de compuestos desconocidos en las áreas siguientes:

- Acidos orgánicos en los fluidos humanos.
- Impurezas en productos químicos.
- Antibióticos.
- Hormonas y ferhormonas de insectos.

Oro en los sistemas Dem Dere

PROSPECTOR suena como un sueño hecho realidad. Muchos informes de sus actuaciones sugieren que, con sólo describirle un terreno, nos dice dónde buscar petróleo, oro o diamantes. Pero, por supuesto, la realidad (aunque excitante de por sí) no es tan mágica como algunas eufóricas noticias de sucesos han sugerido.

A pesar de ello, algunos de los trabajos de PROSPECTOR han sido muy valiosos. Por ejemplo, este sistema experto fue de gran ayuda en el descubrimiento de una ramificación, potencialmente muy rica, de un depósito de molibdeno existente cerca de Mount Tolman en Washington (Winston, 1984). En este caso, los expertos humanos estaban en desacuerdo con los hallazgos de PROSPECTOR, y solamente las perforaciones probaron que el sistema tenía razón.

Alrededor de 500 reglas y más de 300 afirmaciones componen la base de reglas de PROSPECTOR. Sus reglas deductivas trabajan siguiendo las mismas líneas que las reglas de producción de MYCIN (Simons, 1984). No obstante, mientras MYCIN puede combinar bastantes posibilidades (un paciente puede tener más de una infección a la vez), PROSPECTOR utiliza la inferencia probabilística estricta (véase el Apéndice A de este libro relativo al teorema de Bayes), suponiendo que las conclusiones del sistema son excluyentes, es decir, no pueden aparecer dos resultados a la vez (Rich, 1983).

El conocimiento mantenido en PROSPECTOR se obtuvo mediante el típico método de preguntar a expertos humanos (geólogos economistas, en este caso) cómo trabajan, y codificando este saber. Resulta muy interesante el que por separado cada uno de los expertos humanos involucrados dijeran que el proceso de ingeniería del conocimiento les había posibilitado el mejorar sus propias ideas en este campo, al verse forzados a planificar con exactitud lo que hicieron y cómo lo hicieron (Duda, 1981).



3 Creación de sistemas expertos

Hay algunos pasos que son necesarios para el desarrollo de un sistema experto específico. Empezaremos por el obvio de enunciar el problema. Esta declaración se puede hacer en términos exactos, o en términos de lenguaje natural. Posteriormente, necesitamos describir la naturaleza de los datos a introducir que sirven para alcanzar una conclusión (la salida del sistema experto).

Utilizando esta información, es posible encontrar los mecanismos teóricos mediante los cuales se resuelve el problema (es decir, el "cálculo simbólico" que se realizará).

A continuación llegamos a la parte difícil, convertir realmente la percepción del problema, y nuestro deseo para un razonamiento particular mediante el cual se puede hallar la solución, en un procedimiento (o serie de ellos) que funcionen en la práctica. En este punto resulta útil el considerar la incorporación al sistema de un "mecanismo elaborador de informes", que pueda decir al observador humano por qué se han seguido ciertas trayectorias y en definitiva explique cómo ha alcanzado el sistema su conclusión final. Los procedimientos dispondrán, idealmente, de elementos automodificativos, lo que asegura que las trayectorias inútiles se concluyen rápidamente, reservando así las fuerzas y el tiempo del procesado para trayectorias que tienen más probabilidades de estar realmente relacionadas con el problema.

Las dos condiciones fundamentales, lógicamente obvias, que deben ser cumplimentadas para que cualquier sistema experto que se desarrolle resulte útil,

son las que se indican a continuación. La primera es que el sistema debe hacer lo que se pretende hacer. Si hemos realizado un sistema experto para distinguir entre varios tipos de conchas marinas, debe estar capacitado para dar muestras de los adecuados poderes de razonamiento cuando se le confronta con la gran mayoría de los objetos que están dentro de su dominio. En segundo lugar, si un sistema no está escrito de modo que la interacción con el usuario se haga de una forma razonablemente sencilla, será poco probable que se utilice, por muy "inteligente" que tal sistema sea.

Mientras que el dominio de un programa de ordenador para contabilidad personal resulta fácil, la utilización de un sistema experto requiere más que el mero conocimiento de los datos de entrada, y de las salidas previstas. Para utilizarlo adecuadamente, el usuario necesita comprender los límites entre los cuales el programa puede operar.

Es muy probable que el usuario tenga que perder una gran cantidad de tiempo con el programa antes de que él o ella comiencen realmente a comprender sus posibilidades y restricciones. La presentación en pantalla y la documentación del programa deberían servir para reducir tal período en lo posible.

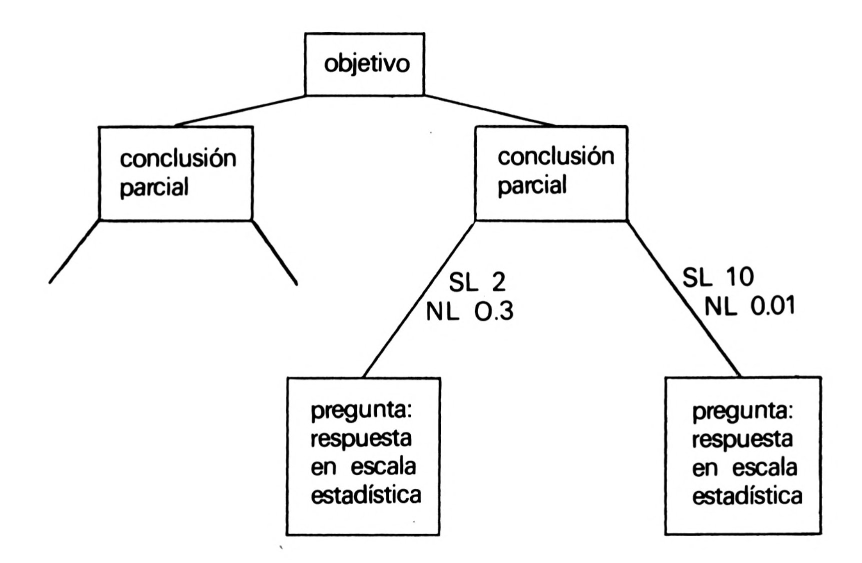
Tipos de información

Normalmente en la mayoría de los sistemas expertos es fácil conocer el dominio general de un programa ("carencias vitamínicas"), el contexto de las entradas ("piel áspera", "tendencia a amoratarse fácilmente") y el de las salidas ("la evidencia indica una marcada insuficiencia de vitamina A en la dieta"). Sin embargo, la determinación del tipo de información con la que el programa es capaz de enfrentarse puede que lleve algún tiempo.

A veces, cuando se está desarrollando un sistema experto, es muy útil el poder decirle que algunas respuestas son más importantes que otras. Imaginemos un programa de diagnóstico médico, dedicado a problemas de espalda. La respuesta a una pregunta como "¿Le duele al paciente cuando intenta levantar un peso?" Bien podría ser más importante que la respuesta a una pregunta como "¿Come el paciente verduras todos los días?" En un caso como éste, el sistema experto necesita un método que le permita enfatizar las respuestas importantes y quitarle importancia a otras accesorias.

Para ello, se utiliza lo que se denominan valores de *suficiencia lógica* para las respuestas que debieran ser sobrevaloradas, y valores de *necesidad lógica* para aquellas otras que debieran ser infravaloradas. Para utilizarlos, *multiplicamos* una respuesta por el valor de suficiencia lógica, si es *verdadera*, y la *dividimos* por el valor de necesidad lógica si es *falsa*. Esto asegura que el sistema tenderá a valorar más a una sola respuesta que tenga importancia que, por ejemplo, a diez que tengan poco interés.

Imaginemos que hemos desarrollado un sistema experto para diagnosis médica que sigue un árbol de decisión, con preguntas y reglas en sus nodos. Este tipo de sistema experto puede partir de enunciados objetivos, formular preguntas adecuadas, y a continuación utilizar sus reglas para tomar decisiones.



Nuestro sistema ha quedado configurado para formular preguntas relevantes ("¿Está sudando mucho el paciente?"), aceptar respuestas como sentencias escogidas de un menú de posibilidades (desde "SI", hasta "NO", pasando por "NO LO SE"), convertirlas en una probabilidad y posteriormente multiplicar las respuestas ciertas por el valor de suficiencia lógica de la pregunta, o dividir las falsas por el valor de necesidad lógica correspondiente.

A menudo es deseable que un sistema experto contenga un mecanismo que detenga ciertas líneas del interrogatorio, si las respuestas específicas dadas anteriormente indican que tales líneas carecen de significado. Continuar después de "¿SE ESTA CAYENDO EL MOTOR A TROZOS POR EL SUELO?" (que como respuesta obtiene, digamos, un SI) con una pregunta como "INTENTE HACER GIRAR EL MOTOR. ¿DA VUELTAS?" es una pérdida de tiempo, se obtienen respuestas que carecen de valor para acercarnos a la conclusión, y demuestran convincentemente la carencia de inteligencia que el sistema realmente posee.

De la misma manera que necesitamos un medio de impedir que ciertas cuestiones se formulen, como resultado de respuestas dadas con anterioridad, necesitamos también ser capaces de formular algunas otras cuestiones, *solamente si* se han recibido respuestas positivas a determinadas preguntas.

Dos tipos de razonamiento

Según parece, existen dos categorías principales en las que las actividades del raciocinio pueden ser emplazadas. El conocimiento de estas categorías simplifica la creación de sistemas expertos específicos, hasta el punto de que es posible (y casi siempre lo es) decidir en cuál de las categorías encajarán la clase particular de problemas a resolver mediante nuestro sistema experto.

Con la idea de mejorar las descripciones, preferimos denominar a estas categorías por razònamiento "exacto" y razonamiento "aproximado". Un problema de lógica exacto es aquel en el que, dadas todas las entradas, hay una salida "verdadera". Un problema aproximado es aquel en el que las entradas están ponderadas numéricamente conforme a su "veracidad", y la salida está también calificada numéricamente.

Como ya sabemos, las dos partes fundamentales de la mayoría de los sistemas expertos (tal como el programa MYCIN de diagnóstico de enfermedades infecciosas, que vimos detalladamente en el capítulo anterior) se denominan *base de conocimientos* (el saber almacenado) y *bloque deductivo* o *motor de inferencia* (el mecanismo de razonamiento). Hemos visto que una gran parte de la información que conocemos acerca del mundo no es susceptible de ser reducida a números. Nuestras vidas están salpicadas de situaciones con frases tales como *parece probable*, *a menudo*, *a veces* y *no sé por qué, pero tengo el presentimiento de que resolverás el problema si...* Para que los sistemas expertos tengan utilidad en el mundo no ideal en el que vivimos, que es incontable numéricamente, deben estar capacitados para trabajar con la percepción y los conceptos humanos.

Los seres humanos no razonan, como norma, solamente (o incluso principalmente) siguiendo reglas. En numerosas ocasiones, el peso dado a ciertos elementos dentro de una proposición lógica se basa en el "conocimiento del mundo real".

Esto se demuestra de forma muy clara en la comprensión de textos hablados o escritos. Las palabras en tales textos tienen solamente significado dentro de un extenso contexto del mundo real. El alcance de nuestro conocimiento del entorno, dentro del cual el texto existe, y al cual refleja y modela, dicta el grado de comprensión que podemos dar al texto. Esta afirmación, por sí misma, nos indica lo dificultoso que es desarrollar sistemas expertos que interactúen con nosotros en lenguaje natural. Con la tecnología actual, es imposible inculcar en una máquina más de una diminuta fracción de la riqueza de nuestro propio conocimiento del mundo real. Y sin esta riqueza, la comprensión del lenguaje natural puede ser solamente efectiva (y no mucho, por cierto) dentro de ámbitos muy reducidos.



4
Sistemas basados en reglas

Resulta bastante fácil crear un sistema experto primitivo basado en reglas. Tales sistemas son quizá de los más sencillos que podemos escribir, aunque pueden ser, no obstante, efectivos y útiles y sirven para darnos una idea de cómo se han creado otros sistemas expertos análogos, pero más complejos, tales como el programa de diagnóstico de enfermedades infecciosas MYCIN.

En este capítulo le presentaremos MECANICO DE AUTOMOVILES, un sistema sencillo basado en reglas, diseñado para ayudar a desesperados profanos en mecánica (como el autor de este libro) a localizar las posibles causas de los problemas que están teniendo con sus coches. Debido a que este programa se basa en información real extraída de expertos humanos en este campo, bien puede ser que le resulte de verdadera utilidad.

No obstante, la razón principal para incluirlos en este libro es la de demostrar cómo se puede desarrollar un sistema basado en reglas en el que éstas están codificadas en un orden específico, y en el que la respuesta a una pregunta dicta la siguiente pregunta que formulará el sistema.

Los dos componentes de una regla

La más sencilla de las reglas utilizadas en un sistema experto tiene dos partes. Estas son: parte izquierda o LHS *(Left Hand Side)*, que es normalmente una sentencia condicional (SI EL ORDENADOR ESTA EN LLAMAS), y una

parte derecha o RHS *(Right Hand Side)*, que normalmente es una cláusula ENTONCES (ENTONCES ROCIALO CON UN EXTINTOR). Un sistema experto basado en reglas de bajo nivel podría ser poco más que una serie de pares LHS-RHS presentados en un orden específico.

Un sistema primitivo de este tipo le permitiría al usuario introducir la respuesta a una pregunta (como, por ejemplo, ¿Está el ordenador en llamas transformándose tranquilamente en una horrible bola negra?) y entonces buscar a través de su colección de reglas para encontrar un emparejamiento entre la respuesta dada y la LHS de una regla. En este punto podría o bien sacar la RHS pertinente o utilizar la RHS para dirigirse a preguntas posteriores más específicas.

El orden en el que se almacena la colección de reglas es importante, ya que es la manera en la que se manejan los emparejamientos LHS. Si hay varias condiciones LHS que deben ser localizadas antes de que una RHS sea evocada, el sistema debe saber cosas tales como cuántos emparejamientos se necesitan y cuán exacta debe ser una respuesta para que constituya un emparejamiento. En algunos sistemas expertos tendrá sentido para evocar una RHS el momento en el que un emparejamiento se ha hallado. En estos sistemas, en los que una única respuesta del usuario puede emparejar con varias LHS, éstas podrían ser dispuestas en orden jerarquizado, con los niveles superiores evocados a ser posible antes de que los de nivel inferior sean incluso investigados. DENDRAL, un sistema basado en reglas que genera información para químicos a partir de datos de espectrometría de masas, funciona con reglas jerarquizadas.

En MECANICO DE AUTOMOVILES, las reglas están en orden estricto, y hay una simple y clara trayectoria hacia la regla siguiente. Esta trayectoria es efectivamente la RHS de la regla.

Veámoslo en acción. Comienza con un menú de posibilidades:

```
ASI QUE TIENES PROBLEMAS CON EL
   COCHE ?
VEAMOS SI PODEMOS SOLUCIONAR EL APURO

PULSA LA TECLA QUE DESCRIBE EL
 PROBLEMA

A.- EL MOTOR NO ANDA, NO FUNCIONA.
B.- EL MOTOR FUNCIONA PERO NO ARRANCA
C.- ARRANCA PERO SE PARA ENSEGUIDA.
D.- EL COCHE ANDA PERO SE ATASCA.
    Y SE CALA FRECUENTEMENTE.
E.- EL COCHE ANDA PERO SIN POTENCIA.
```

Pongamos por ejemplo que nuestro problema es el segundo: EL MOTOR DA VUELTAS PERO NO ARRANCA. Una vez que le hemos dicho esto, nos

conduce a través de una serie de puntos de comprobación y de sugerencias para la realización de reparaciones de emergencia:

```
          < DE ACUERDO, B >
 COMPRUEBA LOS CABLES.
CABE LA POSIBILIDAD DE QUE ESTEN HUME
DOS ?
                (S-SI . N-NO) ?
          < DE ACUERDO, S >
ROCIALOS CON UN SPRAY SUPRESOR DE
HUMEDAD.
 HAY POLVO VISIBLE EN LA PARTE
AISLANTE DE LA BOBINA O EN LA TAPA
DEL DISTRIBUIDOR ?
                (S-SI . N-NO) ?
          < DE ACUERDO, S >
LIMPIA LA PARTE DE LA BOBINA ASI
COMO EL INTERIOR Y EXTERIOR DE LA
TAPA DEL DISTRIBUIDOR.
ASEGURATE DE QUE TODOS LOS CABLES
ESTAN SECOS Y SUJETOS ANTES DE CONTI-
NUAR.
 SI EL MOTOR TODAVIA NO ARRANCA,
ENTONCES HAY QUE MIRAR LAS BUJIAS.
 SALTAN CHISPAS DE 9 O MAS MILIMETROS
ENTRE LOS ELECTRODOS DE LAS BUJIAS ?
                (S-SI . N-NO) ?
```

El orden en el que el sistema experto nos da sus indicaciones y las sugerencias que nos hace dependen de las respuestas que le demos:

```
          < DE ACUERDO, S >
LAS BUJIAS ESTAN DEFECTUOSAS.
 ESTAN SUCIAS LAS BUJIAS ?
                (S-SI . N-NO) ?
          < DE ACUERDO, N >
SI SE TRATA DE UNA EMERGENCIA, PRUEBA
A REDUCIR LA SEPARACION DE LOS ELEC-
TRODOS, MAS O MENOS A LA MITAD.
 TIENES GASOLINA EN EL DEPOSITO ?
                (S-SI . N-NO) ?
          < DE ACUERDO, S >
ESTAN LOS CONDUCTOS DE LA GASOLINA
LIMPIOS ?
                (S-SI . N-NO) ?
```

```
              < DE ACUERDO, S >
RETIRA EL FILTRO DE AIRE DEL CARBURA-
DOR.
 PARECE SECO ?
                    (S-SI . N-NO) ?
              < DE ACUERDO, N >
HAS ESTADO UTILIZANDO MUCHO EL
ARRANQUE EN LOS ULTIMOS MINUTOS ?
                    (S-SI . N-NO) ?
              < DE ACUERDO, S >
ESPERA ALREDEDOR DE UN MINUTO.
SEGUIDAMENTE MANTEN EL ACELERADOR
PISADO A FONDO Y SIN MOVERLO.
 DALE AL CONTACTO, AHORA DEBERIA
ARRANCAR.
                    (S-SI . N-NO) ?
```

Si tiene alguna vez problemas para poner en marcha su coche, y quisiera que su ordenador le librara de tener que llamar a un servicio mecánico de reparaciones de emergencia, cargue el listado siguiente en su ordenador y así tendrá un consejero de MECANICO DE AUTOMOVILES a su disposición:

```
10 REM MECANICO DE AUTOMOVILES
20 CLS:KEYOFF
30 PRINT "ASI QUE TIENES PROBLEMAS CO
N EL         COCHE ?"
40 PRINT "VEAMOS SI PODEMOS SOLUCIONA
R EL APURO"
50 PRINT
60 PRINT "PULSA LA TECLA QUE DESCRIBE
 EL          PROBLEMA"
70 PRINT
80 PRINT "A.- EL MOTOR NO ANDA, NO FU
NCIONA."
90 PRINT "B.- EL MOTOR FUNCIONA PERO
NO ARRANCA"
100 PRINT "C.- ARRANCA PERO SE PARA E
NSEGUIDA."
110 PRINT "D.- EL COCHE ANDA PERO SE
ATASCA."
120 PRINT "     Y SE CALA FRECUENTEMEN
TE."
```

```
130 PRINT "E.- EL COCHE ANDA PERO SIN
 POTENCIA."
140 IF INKEY$="" THEN 140
150 A$=INKEY$
160 IF A$<"A" OR A$>"E" THEN 150
170 GOSUB 1540:CLS
180 ON ASC(A$)-64 GOTO 210,580,1180,1
210,1330
190 END
200 REM ****************************
210 REM MOTOR NO ANDA
220 PRINT:PRINT "EMPECEMOS COMPROBAND
O LA BATERIA."
230 PRINT "ENCIENDE LAS LUCES DEL COC
HE."
240 PRINT TAB(6);"TIENEN POCA INTENSI
DAD ?"
250 GOSUB 1500
260 IF A$="N" THEN 370:REM BATERIA EN
 BUEN ESTADO
270 PRINT "SE HA SOLTADO ALGUN CABLE
DE LA"
280 PRINT "BATERIA O ESTAN SUS TERMIN
ALES        SULFATADOS ?"
290 GOSUB 1500
300 IF A$="S" THEN PRINT "LIMPIALOS Y
 APRIETALOS FUERTE."
310 GOSUB 1560
320 PRINT "ESTA FLOJA LA CORREA DEL V
ENTILADOR ?"
330 GOSUB 1500
340 IF A$="S" THEN PRINT TAB(5);"ASEG
URA Y TENSA LA CORREA."
350 GOSUB 1560
360 PRINT "PONTE AL VOLANTE Y PRUEBA;
 SI NO      DANDOLE UN EMPUJON DEBERIA
 ARRANCAR.":GOTO 1590
370 PRINT "ESTA AFLOJADO O SUCIO EL E
XTREMO DEL CABLE QUE VA DE LA BATERIA
"
380 PRINT "AL MOTOR DE ARRANQUE ?"
390 GOSUB 1500
400 IF A$="S" THEN PRINT "APRIETALO Y
 LIMPIA LAS CONEXIONES."
410 GOSUB 1560
```

```
420 PRINT "PUENTEA EL CONTACTO DEL MO
TOR DE     ARRANQUE."
430 PRINT "FUNCIONA AHORA ?"
440 GOSUB 1500
450 IF A$="S" THEN PRINT "PROBABLEMEN
TE LA LLAVE DE CONTACTO NO FUNCIONA C
ORRECTAMENTE."
460 GOSUB 1560
470 IF A$="S" THEN PRINT "DEBERIAS CA
MBIARLA.":GOTO 1590
480 PRINT "HACE UN CHASQUIDO EL ARRAN
QUE?"
490 GOSUB 1500
500 IF A$="S" THEN PRINT "PUEDE SER Q
UE EL MOTOR DE ARRANQUE   ESTE ATASCA
DO.":GOTO 550
510 PRINT "SI NO LO HACE, PROBABLEMEN
TE SE DEBE AL CONTACTOR."
520 PRINT "UN EMPUJON PUEDE HACER QUE
 ARRANQUE.":GOTO 1590
530 REM ** STARTER ATASCADO **
540 PRINT " QUITA EL CONTACTO ."
550 PRINT " METE UNA VELOCIDAD LARGA
Y EMPUJA"
560 PRINT "EL COCHE UNOS CENTIMETROS
PARA DESA- TASCAR EL ARRANQUE.":GOTO
1590
570 REM *************************
580 REM DA VUELTAS PERO NO ARRANCA
590 PRINT " COMPRUEBA LOS CABLES."
600 PRINT "CABE LA POSIBILIDAD DE QUE
 ESTEN HUMEDOS ?"
610 GOSUB 1500
620 IF A$="S" THEN PRINT "ROCIALOS CO
N UN SPRAY SUPRESOR DE    HUMEDAD.":G
OSUB 1560
630 PRINT " HAY POLVO VISIBLE EN LA P
ARTE        AISLANTE DE LA BOBINA O EN
 LA TAPA"
640 PRINT "DEL DISTRIBUIDOR ?"
650 GOSUB 1500
660 IF A$="S" THEN PRINT "LIMPIA LA P
ARTE DE LA BOBINA ASI"
670 IF A$="S" THEN PRINT "COMO EL INT
ERIOR Y EXTERIOR DE LA    TAPA DEL DI
```

```
STRIBUIDOR.":GOSUB 1560
680 PRINT "ASEGURATE DE QUE TODOS LOS
 CABLES    ESTAN SECOS Y SUJETOS ANTE
S DE CONTI-NUAR."
690 GOSUB 1560
700 PRINT " SI EL MOTOR TODAVIA NO AR
RANCA,      ENTONCES HAY QUE MIRAR LAS
 BUJIAS."
710 PRINT " SALTAN CHISPAS DE 9 O MAS
 MILIMETROS"
720 PRINT "ENTRE LOS ELECTRODOS DE LA
S BUJIAS ?"
730 GOSUB 1500
740 IF A$="S" THEN PRINT "LAS BUJIAS
ESTAN DEFECTUOSAS.":GOSUB 1560
750 IF A$="N" THEN 820
760 PRINT " ESTAN SUCIAS LAS BUJIAS ?
"
770 GOSUB 1500
780 IF A$="S" THEN PRINT " NO SE PUED
E HACER UNA REPARACION DE EMERGENCIA"
790 IF A$="S" THEN PRINT " MIENTRAS L
AS BUJIAS ESTEN ASI.":GOSUB 1560:GOTO
 760
800 IF A$="N" THEN PRINT "SI SE TRATA
 DE UNA EMERGENCIA, PRUEBA"
810 IF A$="N" THEN PRINT "A REDUCIR L
A SEPARACION DE LOS ELEC- TRODOS, MAS
 O MENOS A LA MITAD.":GOSUB 1560:GOTO
 900
820 PRINT " NO SALTA NINGUNA CHISPA ?
"
830 GOSUB 1500
840 IF A$="N" THEN 760
850 GOSUB 860:GOTO 1590
860 PRINT "OBSERVA SI HAY ALGUNA GRIE
TA EN EL   ROTOR, LA BOBINA O LA TAPA
 DEL DISTRIBUIDOR."
870 PRINT "  SI NO HAY NINGUNA,"
880 PRINT " ENTONCES EL PROBLEMA PUED
E QUE ESTE EN LOS PLATINOS O EN EL CO
NDENSADOR."
890 PRINT " UNA REPARACION PUEDE SER
NECESARIA.":RETURN
900 PRINT " TIENES GASOLINA EN EL DEP
```

```
OSITO ?"
910 GOSUB 1500
920 IF A$="N" THEN PRINT "LLENALO E I
NTENTALO OTRA VEZ.":GOSUB 1560
930 PRINT "ESTAN LOS CONDUCTOS DE LA
GASOLINA   LIMPIOS ?"
940 GOSUB 1500
950 IF A$="N" THEN PRINT "ATIENDELOS
Y PRUEBA OTRA VEZ.":GOSUB 1560
960 PRINT "RETIRA EL FILTRO DE AIRE D
EL CARBURA-DOR."
970 GOSUB 1560
980 PRINT " PARECE SECO ?"
990 GOSUB 1500
1000 IF A$="N" THEN 1070
1010 PRINT " GIRA EL MOTOR UNAS CUANT
AS VECES,"
1020 PRINT "BLOQUEANDO CON LA MANO LA
 TOMA DE    AIRE.":GOSUB 1560
1030 PRINT "ESTA TU MANO HUMEDA DE GA
SOLINA ?"
1040 GOSUB 1500
1050 IF A$="N" THEN PRINT "DESENROSCA
 LA CUBA DE GASOLINA; EN EL CASO DE Q
UE EL RESPIRADERO ESTE OBS-TRUIDO,"
1060 IF A$="N" THEN PRINT "PUEDE QUE
NO FUNCIONE LA BOMBA DE LA GASOLINA."
:GOSUB 890:GOTO 1590
1070 PRINT "HAS ESTADO UTILIZANDO MUC
HO EL        ARRANQUE EN LOS ULTIMOS M
INUTOS ?"
1080 GOSUB 1500
1090 IF A$="N" THEN 1130
1100 PRINT "ESPERA ALREDEDOR DE UN MI
NUTO.        SEGUIDAMENTE MANTEN EL AC
ELERADOR"
1110 PRINT "PISADO A FONDO Y SIN MOVE
RLO."
1120 PRINT " DALE AL CONTACTO, AHORA
DEBERIA      ARRANCAR.":GOTO 1590
1130 PRINT " PUEDE QUE LA VALVULA DEL
 FLOTADOR   ESTE AGARROTADA, QUEDANDO
SE ABIERTA."
1140 PRINT " Y QUE EL CARBURADOR ESTE
 INUNDADO.":GOSUB 1560
```

```
1150 PRINT "GOLPEA LIGERAMENTE LA TAP
A DEL CARBU-RADOR PARA QUE SE CIERRE.
"
1160 PRINT "DESPUES INTENTA ARRANCAR
OTRA VEZ.":GOTO 1590
1170 REM ***************************
1180 REM  ARRANCA PERO SE PARA
1190 PRINT "ESTO SUGIERE UN FUNCIONAM
IENTO DEFEC-TUOSO DE LA RESISTENCIA T
ERMICA, QUE DEBERIA SER SUSTITUIDA.":
GOTO 1590
1200 REM ***************************
1210 REM FUNCIONA PERO SE CALA MUCHO
1220 PRINT "LA CAUSA PUEDE SER CUALQU
IERA DE LAS SIGUIENTES :"
1230 PRINT "- CABLES SUELTOS O EN COR
TOCIRCUITO."
1240 PRINT "- CHISPAS DEBILES."
1250 PRINT "- UN FALLO EN EL SISTEMA
DE ALIMENTA-  CION."
1260 PRINT:PRINT "COMPRUEBA PRIMERO Q
UE NO HAY NINGUN  CABLE SUELTO O EN C
ORTOCIRCUITO."
1270 PRINT "SI NO ESTAN BIEN, REPARAL
OS.  EN CASO CONTRARIO, EL FALLO PODR
IA ESTAR EN LAS BUJIAS."
1280 GOSUB 860
1290 PRINT " PODEMOS HACER UNA ULTIMA
 COMPROBA-"
1300 PRINT "CION; LOS PLATINOS.":GOSU
B 1560
1310 GOTO 1340
1320 REM ***************************
1330 REM ANDA, PERO HOLGAZANEANDO
1340 PRINT "BIEN PODRIA SER QUE UNA O
 VARIAS DE"
1350 PRINT "LAS BUJIAS, ESTEN DEFECTU
OSAS.":GOSUB 1560
1360 PRINT " DESCONECTA UNA CADA VEZ;
"
1370 PRINT " AQUELLAS CUYA DESCONEXIO
N NO AFECTE AL RALENTI, SON LAS QUE F
ALLAN."
1380 PRINT "COMPRUEBALAS VOLVIENDO AL
 SISTEMA."
```

```
1390 GOSUB 1560
1400 PRINT "COMO RESULTADO DE LA COMP
ROBACION,   HAS ENCONTRADO ALGUNA BUJ
IA DEFEC-   TUOSA ?"
1410 GOSUB 1500
1420 IF A$="S" THEN PRINT "A SER POSI
BLE CAMBIALAS TODAS, O AL"
1430 IF A$="S" THEN PRINT "MENOS LAS
DEFECTUOSAS.":GOSUB 1560
1440 PRINT "DESCARTADO EL FALLO DE BU
JIAS, LA"
1450 PRINT " CAUSA MAS FRECUENTE DE U
N RALENTI   IRREGULAR ES UNA MEZCLA D
EMASIADO "
1460 PRINT "RICA; POR LO QUE DEBERIAS
 AJUSTARLA."
1470 PRINT "PRUEBA TAMBIEN A AJUSTAR
EL RALENTI."
1480 GOTO 1590
1490 REM **************************
1500 PRINT TAB(16);"(S-SI . N-NO) ?"
1510 IF INKEY$<>"" THEN 1510
1520 A$=INKEY$
1530 IF A$<>"S" AND A$<>"N" THEN 1520
1540 PRINT TAB(10);"< DE ACUERDO, ";A
$;" >"
1550 BEEP:REM INCLUYE 'BEEP' O UN
              SONIDO SIMILAR EN ESTA
              LINEA
1560 FOR T=0 TO 1000:NEXT T
1570 PRINT
1580 RETURN
1590 PRINT
1600 PRINT "HAS ENCONTRADO YA LA SOLU
CION ?"
1610 GOSUB 1500
1620 IF A$="S" THEN CLS:END
1630 GOTO 20
```

Espero que al utilizar este programa, además de ver cómo puede funcionar un sistema basado en reglas, pueda también apreciar las desventajas reales de tener un sistema en el que las reglas son *hard-wired.* Estas resultan extremadamente difíciles de modificar, so pena de destruir casi totalmente el programa. También pudiera resultar difícil la localización de un error de apreciación

mediante el sistema, y no hay posibilidad de que éste "aprenda su tarea" mientras trabaja, tal y como lo hacen otros programas (como FUZZY RITA) que veremos después en este mismo libro.

A pesar de sus desventajas, un sistema *hard-wired* construido para un propósito concreto, como MECANICO DE AUTOMOVILES, es mucho más práctico dentro de su dominio de lo que sería un sistema general que hubiera sido enseñado en relación al tema.

Al determinar el tipo de sistema experto que mejor solucione unas necesidades específicas, será necesario ponderar estas ventajas y desventajas.



5
El doctor está dentro

En mi libro *Inteligencia artificial: conceptos y programas* (Anaya Multimedia, 1985) presenté un sistema experto imaginario llamado MEDICO, que hacía una serie de preguntas y a partir de las respuestas daba un informe sobre salud y longevidad. Ahora he decidido crearlo realmente para este libro.

El mejor camino para vivir muchos años es elegir padres y abuelos longevos. Como esta elección está fuera del alcance de todos nosotros, lo mejor es aceptar el segundo camino que consiste en vivir día a día aplicando los conocimientos sobre salud y nutrición que el hombre ha adquirido. Por supuesto, pocos de nosotros vivimos lo inteligentemente que podríamos hacerlo, o aplicamos toda la información de que disponemos. Si no fuera así, la vida sería tal vez bastante aburrida.

MEDICI le permitirá deducir cómo vive realmente, en vez de pedirle información sobre la salud en la que usted cree, pero no aplica coherentemente a su estilo de vida.

Una vez que el sistema experto le haya dado sus "condiciones" de salud y comentado su estado, le indicará el rango dentro del que puede situarse su esperanza de vida. Entonces, puede volver a hacer funcionar el sistema, y ver qué cambios debería introducir en su estilo de vida para poder incrementar el número de años durante los cuales podría escribir sistemas expertos en su ordenador.

Probemos MEDICI con alguien (por ejemplo, alguien con pocas ganas de trabajar) y veamos lo que ocurre:

CUAL DE LAS SIGUIENTES AFIRMACIONES
ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

A - MI PESO ES REALMENTE EXCESIVO.
B - MI PESO ES MAYOR DE LO NORMAL.
C - MI PESO ES LIGERAMENTE ELEVADO.
D - MI PESO ES LO JUSTO PARA MI.
E - SOY MAS DELGADO DE LO QUE DESEO.

DE ACUERDO ;C
-5

CUAL DE LAS SIGUIENTES AFIRMACIONES
ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

ME OCUPO DE HACER EJERCICIO QUE
ELEVE MIS PULSACIONES A 120 O MAS,
DURANTE AL MENOS EL SIGUIENTE NUMERO
DE HORAS A LA SEMANA:

A - MENOS DE UN CUARTO DE HORA.
B - ENTRE UNO Y TRES CUARTOS DE HORA
C - ENTRE TRES CUARTOS Y HORA Y
MEDIA.
D - ENTRE UNA Y MEDIA Y DOS Y MEDIA.
E - MAS DE DOS Y MEDIA.

DE ACUERDO ;A
O

CUAL DE LAS SIGUIENTES AFIRMACIONES
ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

CUANDO CONDUZCO :
A - MUY RARAS VECES LLEVO EL CINTURON
DE SEGURIDAD.
B - LA CUARTA PARTE DE LAS VECES
LLEVO PUESTO EL CINTURON DE SEGURIDAD
.

C - DE CADA DOS DESPLAZAMIENTOS, EN UNO LLEVO PUESTO EL CINTURON DE SEGURIDAD.
D - LA MAYOR PARTE DE LAS VECES ME PONGO EL CINTURON, PERO NO SIEMPRE.
E - SIEMPRE UTILIZO EL CINTURON.

DE ACUERDO ;E

8

CUAL DE LAS SIGUIENTES AFIRMACIONES ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

SOY CONSCIENTE DE LA NUTRICION Y TRATO DE COMER CON SALUD ...

A - SIEMPRE.
B - CASI SIEMPRE.
C - LA MITAD DE LAS VECES.
D - DE VEZ EN CUANDO.
E - PRACTICAMENTE NUNCA.

DE ACUERDO ;D

1

CUAL DE LAS SIGUIENTES AFIRMACIONES ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

TABACO (LOS PUROS Y CIGARROS CUENTAN COMO CIGARRILLOS)
A - NO FUMO.
B - MENOS DE 15 CIGARRILLOS AL DIA.
C - DE 15 A 25 CIGARRILLOS AL DIA.
D - DE 26 A 42 CIGARRILLOS AL DIA.
E - MAS DE 42 CIGARRILLOS AL DIA.

DE ACUERDO ;A

0

CUAL DE LAS SIGUIENTES AFIRMACIONES ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

ALCOHOL - CUANTAS BEBIDAS (POR TERMINO MEDIO) TOMAS CADA DIA ?

A - NINGUNA.
B - MENOS DE TRES.
C - ENTRE 3 Y 6 .
D - ENTRE 7 Y 9 .
E - MAS DE 9 .

DE ACUERDO ;C
-6

CUAL DE LAS SIGUIENTES AFIRMACIONES ESTA MAS CERCA DE LA VERDAD ?
(SELECCIONA UNA)

EN GENERAL, COMO HA SIDO TU VIDA EN LOS ULTIMOS 6 MESES ; SEGUN TU OPINION ?

A - EXTREMADAMENTE AJETREADA.
B - MEDIANAMENTE AJETREADA.
C - LIGERAMENTE AJETREADA.
D - NORMAL.
E - NADA AJETREADA, MUY TRANQUILA.

DE ACUERDO ;C
-5

VALORACION PERSONAL DE MEDICI :

PESO :-5
EJERCICIO : 0
AUTO-SEGURIDAD : 8
NUTRICION : 1
TABACO : 0
ALCOHOL :-6
STRESS :-5
LA VALORACION TOTAL ES :-7

```
EN UNA ESCALA DONDE CERO ES EL
TERMINO MEDIO, EL MINIMO SE ALCANZA
EN -83, Y EL MAXIMO EN 37.

 ESTO INDICA QUE TU ESTADO DE SALUD
ESTA POR DEBAJO DEL TERMINO MEDIO.

    ESPERANZA DE VIDA :
    HOMBRE         MUJER
    60 A 66        65 A 71
```

A continuación viene el listado, para que pueda comprobar su estado actual de salud:

```
10 REM * MEDICI - CHEQUEO PERSONAL *
20 REM       TODAS LAS ENTRADAS
                EN MAYUSCULAS
30 CLS:KEYOFF
40 GOSUB 1260
50 PRINT:PRINT"A - MI PESO ES REALMEN
TE EXCESIVO."
60 PRINT:PRINT"B - MI PESO ES MAYOR D
E LO NORMAL."
70 PRINT:PRINT"C - MI PESO ES LIGERAM
ENTE ELEVADO."
80 PRINT:PRINT"D - MI PESO ES LO JUST
O PARA MI."
90 PRINT:PRINT"E - SOY MAS DELGADO DE
 LO QUE DESEO."
100 GOSUB 1150
110 PESO=5*(ASC(A$)-68):IF A$="E" THE
N PESO=0
120 PRINT PESO
130 GOSUB 1230
140 PRINT " ME OCUPO DE HACER EJERCIC
IO  QUE"
150 PRINT "ELEVE MIS PULSACIONES A 12
0 O MAS,"
160 PRINT "DURANTE AL MENOS EL SIGUIE
NTE NUMERO"
170 PRINT " DE HORAS A LA SEMANA:"
180 PRINT
```

```
190 PRINT:PRINT "A - MENOS DE UN CUAR
TO DE HORA."
200 PRINT:PRINT"B - ENTRE UNO Y TRES
CUARTOS DE HORA"
210 PRINT:PRINT "C - ENTRE TRES CUART
OS Y HORA  Y          MEDIA."
220 PRINT "D - ENTRE UNA Y MEDIA Y DO
S Y MEDIA."
230 PRINT:PRINT "E - MAS DE DOS Y MED
IA."
240 GOSUB 1150
250 EJERCICIO=5*(ASC(A$)-63)-5:IF A$=
"A" THEN EJERCICIO=0
260 PRINT EJERCICIO
270 GOSUB 1230
280 PRINT " CUANDO CONDUZCO :":PRINT
290 PRINT:PRINT "A - MUY RARAS VECES
LLEVO EL CINTURON DE SEGURIDAD."
300 PRINT:PRINT "B - LA CUARTA PARTE
DE LAS VECES     LLEVO PUESTO EL CINT
URON DE SEGURIDAD."
310 PRINT:PRINT "C - DE CADA DOS DESP
LAZAMIENTOS, EN  UNO LLEVO PUESTO EL
CINTURON DE SEGU-RIDAD."
320 PRINT:PRINT "D - LA MAYOR PARTE D
E LAS VECES ME   PONGO EL CINTURON, P
ERO NO SIEMPRE."
330 PRINT:PRINT "E - SIEMPRE UTILIZO
EL CINTURON."
340 GOSUB 1150
350 CI=2*(ASC(A$)-65)
360 PRINT CI
370 GOSUB 1230
380 PRINT " SOY CONSCIENTE DE LA NUTR
ICION Y    TRATO DE COMER CON SALUD .
.."
390 PRINT
400 PRINT:PRINT "A - SIEMPRE."
410 PRINT:PRINT "B - CASI SIEMPRE."
420 PRINT:PRINT "C - LA MITAD DE LAS
VECES."
430 PRINT:PRINT "D - DE VEZ EN CUANDO
."
440 PRINT:PRINT "E - PRACTICAMENTE NU
NCA."
```

```
450 GOSUB 1150
460 DIETA=-ASC(A$)+69
470 PRINT DIETA:GOSUB 1230
480 PRINT "TABACO ( LOS PUROS Y CIGAR
ROS CUENTAN COMO CIGARRILLOS )"
490 PRINT:PRINT "A - NO FUMO."
500 PRINT:PRINT "B - MENOS DE 15 CIGA
RRILLOS AL DIA."
510 PRINT:PRINT "C - DE 15 A 25 CIGAR
RILLOS AL DIA."
520 PRINT:PRINT "D - DE 26 A 42 CIGAR
RILLOS AL DIA."
530 PRINT:PRINT "E - MAS DE 42 CIGARR
ILLOS AL DIA."
540 GOSUB 1150
550 TABACO=-7*(ASC(A$)-65)
560 PRINT TABACO
570 GOSUB 1230
580 PRINT " ALCOHOL - CUANTAS BEBIDAS
 (POR TER- MINO MEDIO) TOMAS CADA DIA
 ?"
590 PRINT
600 PRINT:PRINT "A - NINGUNA."
610 PRINT:PRINT "B - MENOS DE TRES."
620 PRINT:PRINT "C - ENTRE 3 Y 6 ."
630 PRINT:PRINT "D - ENTRE 7 Y 9 ."
640 PRINT:PRINT "E - MAS DE 9 ."
650 GOSUB 1150
660 BEBIDA=-30
670 IF A$="A" THEN BEBIDA=0
680 IF A$="B" THEN BEBIDA=1
690 IF A$="C" THEN BEBIDA=BEBIDA/5
700 IF A$="D" THEN BEBIDA=BEBIDA/2
710 PRINT BEBIDA
720 GOSUB 1230
730 PRINT " EN GENERAL, COMO HA SIDO
TU VIDA EN"
740 PRINT " LOS ULTIMOS 6 MESES ; SEG
UN TU      OPINION ?"
750 PRINT
760 PRINT:PRINT "A - EXTREMADAMENTE A
JETREADA."
770 PRINT:PRINT "B - MEDIANAMENTE AJE
TREADA."
780 PRINT:PRINT "C - LIGERAMENTE AJET
```

```
READA."
790 PRINT:PRINT "D - NORMAL."
800 PRINT:PRINT "E - NADA AJETREADA,
MUY TRANQUILA."
810 GOSUB 1150
820 STRESS=INT(2.5*(ASC(A$)-69))
830 PRINT STRESS
840 GOSUB 1290
850 PRINT:PRINT " VALORACION PERSONAL
 DE MEDICI :"
860 PRINT
870 PRINT TAB(5);"PESO :";PESO
880 PRINT TAB(5);"EJERCICIO :";EJERCI
CIO
890 PRINT TAB(5);"AUTO-SEGURIDAD :";C
I
900 PRINT TAB(5);"NUTRICION :";DIETA
910 PRINT TAB(5);"TABACO :";TABACO
920 PRINT TAB(5);"ALCOHOL :";BEBIDA
930 PRINT TAB(5);"STRESS :";STRESS
940 GOSUB 1290
950 ANT=PESO+EJ+CI+DI+TA+BE+ST
960 GOSUB 1290:PRINT
970 PRINT "  LA VALORACION TOTAL ES :
";ANT:PRINT
980 PRINT "EN UNA ESCALA DONDE CERO E
S EL"
990 PRINT "TERMINO MEDIO, EL MINIMO S
E ALCANZA"
1000 PRINT "EN -83, Y EL MAXIMO EN 37
 ."
1010 GOSUB 1290
1020 PRINT:IF ANT<6 AND ANT>-6 THEN A
$="TA EN UN TERMINO MEDIO.":L$="62 A
73  72 A 78"
1030 IF ANT<-5 AND ANT>-21 THEN A$="T
A POR DEBAJO DEL TERMINO MEDIO.":L$="
60 A 66  65 A 71"
1040 IF ANT<-20 THEN A$=" POBRE.":L$=
"60 O MENOS  65 O MENOS"
1050 IF ANT<-45 THEN A$=" MUY POBRE."
1060 IF ANT<-45 THEN A$=" MUY, MUY PO
BRE."
1070 IF ANT>5 AND ANT<15 THEN A$=" BU
ENO.":L$="74 A 80  79 A 85"
```

```
1080 IF ANT>14 THEN A$=" ESTUPENDAMEN
TE BUENO.":L$="81 O MAS  86 O MAS"
1090 PRINT " ESTO INDICA QUE TU ESTAD
O DE SALUD  ES";A$
1100 PRINT
1110 PRINT "    ESPERANZA DE VIDA :"
1120 PRINT TAB(4);"HOMBRE          MUJE
R"
1130 PRINT TAB(4);L$
1140 PRINT:END
1150 REM ***********************
1160 REM ACEPTACION DE ENTRADAS
1170 IF INKEY$<>"" THEN 1170
1180 A$=INKEY$
1190 IF A$<"A" OR A$>"E" THEN 1180
1200 PRINT :PRINT TAB(12);"DE ACUERDO
 ;";A$
1210 RETURN
1220 REM **********************
1230 REM RETARDO // PREGUNTA
1240 FOR J=1 TO 1000:NEXT J
1250 CLS:CLS
1260 PRINT:PRINT
1270 PRINT " CUAL DE LAS SIGUIENTES A
FIRMACIONES ESTA MAS CERCA DE LA VERD
AD ?             ( SELECCIONA UNA )"
1280 PRINT
1290 FOR J=1 TO 400:NEXT J
1300 RETURN
```

Nota.—El conocimiento de MEDICI proviene de haber realizado una gran cantidad de aproximaciones sucesivas utilizando la información de: *Social Readjustment Rating Scale*, Metropolitan Life Insurance Company, Dr. Thomas Holmes. *Life Plan for your Health*, Addison-Wesley, Reading MA., 1978, Dr. Donald Vickey. *New Age Training for Fitness and Health*, Grove House, Inc., San Francisco, CA., 1979, Dyveke Spino. Teniendo en cuenta que yo no soy un médico, que he realizado muchas aproximaciones toscas pero eficaces a la hora de trabajar con el programa y la gran cantidad de veces que lo he ejecutado hasta conseguir coincidencias y aproximaciones con los resultados, sus salidas deben ser entendidas dentro del contexto informativo en el que nos estamos moviendo. Sin embargo, no estarán totalmente alejadas de la realidad y pueden servir para demostrar lo que un sistema experto puede hacer en el campo del diagnóstico.



6

Desarrollo de su propio sistema experto

A pesar de que los programas del tipo de MECANICO DE AUTOMOVILES y MEDICI son interesantes y demuestran con claridad cómo funcionan muchos de los sistemas expertos actualmente utilizados en el mundo, adolecen de dos grandes desventajas. La primera es que sólo codifican el saber en algún tema específico, es decir, tienen un dominio acotado. La segunda desventaja es que, antes de que el programa pueda escribirse, alguien tiene que poseer realmente el saber y conocer las respuestas, a fin de construir la base de conocimientos.

Si usted ya sabe cómo comprobar las pequeñas averías de su coche, probablemente no encontrará interesante el conectar su ordenador para descubrir el estado de sus bujías.

No obstante, como ya mencioné antes, muchos sistemas expertos del "mundo real" solamente se desarrollaron después de que se hubiera extraído el saber de un experto humano poniéndolo en una especie de estructura sistemática, por un "ingeniero del conocimiento". El ingeniero intenta descubrir la heurística mediante la cual el experto humano resuelve un problema. (La heurística es un camino hacia un objetivo que ha sido resuelto por la experiencia, en vez de mediante cálculos; es un trabajo empírico. No se garantiza que un camino de este tipo obtenga un resultado seguro, aunque la experiencia ha demostrado que en la mayoría de los casos es muy probable el conseguir acercarse a la consecución de un objetivo particular. Los programas de ajedrez juegan, en gran parte, de forma heurística. Un enfoque "algorítmico", en contraste con uno

heurístico, es aquel en el que se aplica una técnica o procedimiento que produce inevitablemente un resultado particular. Su ordenador utiliza algoritmos residentes en su interior para cosas como la adición de números, etc.)

El saber organizado procedente de un experto humano se preserva en el interior de un programa, de modo que pueda ser grabado. Pensemos en lo interesante y valioso que sería si pudiéramos crear un programa que estuviera capacitado para deducir un conjunto de reglas, por sí mismo, partiendo simplemente de los datos en bruto. Esto sería un gran logro, y aún lo sería mucho más si pudiéramos escribir un programa que hiciera lo anterior en un campo en el que incluso los expertos humanos carecieran de reglas precisas.

Si esto fuera posible —hacer un programa que pudiera crear sus propias reglas partiendo de datos en bruto, y utilizarlos para obtener resultados satisfactorios en campos en los que los humanos no pudieran—, sería fenómeno conseguir que el programa pudiera explicar la utilización de la esencia de las reglas que ha desarrollado, de forma que pudiéramos aplicarlas nosotros mismos. Sería fenómeno, ya que en este caso, además de ser un sirviente, el sistema experto se convertiría entonces en un ayudante para la investigación y en un instructor.

En esta sección del libro vamos a describir un programa con estas características. Aunque puede que los estadísticos se asusten un poco de la manera poco ortodoxa en la que el programa manipula a veces los números, la prueba de la validez de un sistema experto está en sus resultados. Si es capaz de convertirse en un experto en casi cualquier cosa, incluso en campos en los que los seres humanos son incapaces de crear de manera heurística fiable, realmente no importa demasiado lo que pase con los números.

Encuentro con FUZZY RITA

Nuestro sistema tiene otra característica. Muchas de las decisiones que acometen los expertos del mundo real no tienen resultados del tipo blanco-negro. Aunque la respuesta a la pregunta "¿Es el número A mayor que el número B?" puede tener sólo una respuesta, una pregunta como "¿Es buena para usted la vitamina X?" nos traslada rápidamente al mundo del "depende", "la vitamina X es a menudo buena para los niños recién nacidos, excepto cuando tienen sangre del tipo ES; si sus madres practican karate el valor X puede variar para tales niños". Esta "regla empírica" incluye *a menudo buena*, *excepto cuando* y *puede variar*; conceptos bastante vagos y confusos.

Como habrá usted adivinado, FUZZY RITA está familiarizada con un mundo en el que la causa y el efecto pueden no estar siempre relacionados de forma directa. (El nombre *Rita* viene de la famosa comedia del London West End, *Educating Rita.* Tal como indica la subrutina de la línea 970, nuestro programa puede ser educado. El adjetivo *Fuzzy* (difuso) no sólo es un nombre bonito, sino que hace referencia a la naturaleza del razonamiento que este programa puede desarrollar. No obstante, también funcionará bien en un mundo blanco y negro de respuestas sí/no.)

Los componentes de un sistema experto

Antes de que procedamos con el desarrollo de nuestro sistema, veremos otra vez lo que discutimos antes al mirar las partes principales de los sistemas expertos en general. La mayor parte de los programas disponibles en el comercio tienen dos componentes principales: un *bloque de deducción* y una *base de conocimientos*. El primero de ellos es el mecanismo mediante el cual el sistema experto alcanza sus conclusiones. La base de conocimientos, obviamente, son los datos estrictos que el sistema manipula con el fin de alcanzar tales conclusiones.

El mantener las dos partes del sistema separadas tiene una gran ventaja. Si el mecanismo de inferencia es un dispositivo de propósito general, debería poder trabajar con bases de conocimientos de varios dominios, "pensando" en cualquier campo determinado, por la naturaleza de la base dada.

Vimos esto anteriormente al hablar de MYCIN, el potente sistema de Stanford para diagnosis de enfermedades infecciosas. Al despojarlo de su base de conocimientos, se convirtió en el bloque de deducción EMYCIN (de *Essential MYCIN)*. La adición de nuevas bases de conocimientos transformó al "doctor" primero en un MECANICO DE AUTOMOVILES (con la ayuda de un *Pontiac Service Manual* de 1975), después en un consejero de análisis estructural SACON, posteriormente en un especialista médico en el campo de los problemas sanguíneos (CLOT), y también en sistemas con nombres tan intrigantes como LITHO, HEADMED y BLUEBOX.

FUZZY RITA no es de la misma familia que MYCIN y sus descendientes. En vez de haber tenido que suministrarle un montón de reglas, FUZZY RITA desarrolla las suyas propias, modificándolas cuando es necesario si sus conclusiones son equivocadas. Resulta fascinante observar cómo se va construyendo una regla base; FUZZY RITA nos permite la posibilidad de observar el estado actual de las reglas después de haber tomado cada decisión.

Un sistema de uso general

FUZZY RITA fue creada específicamente para ser un sistema general, capacitado para trabajar con los datos de casi cualquier problema. Esto implica que no siempre es tan inteligente como podría. Así, puede que resulte a veces un poco prolija solicitando información antes de alcanzar una conclusión (aunque tiene la habilidad, cuando hay más de dos conclusiones posibles, de determinar que ciertas preguntas no pueden afectar al resultado, y, en tales casos, simplemente se las salta).

Por tanto, si desea utilizar RITA en un campo específico, por un largo período de tiempo, es conveniente "pellizcarlo" un poco, una vez que lo tenga establecido y en funcionamiento, con el fin de mejorar su rendimiento. La valoración que da el sistema a las coincidencias totales entre las respuestas del usuario y la información de su base de conocimientos adquirida, y la atención que presta a aquellas respuestas que no son totalmente correctas, pueden

parecer insignificantes en orden a conseguir los mejores resultados. Todos los sistemas expertos se comprueban introduciéndoles ejemplos conocidos, y comparando las respuestas del sistema con los resultados conocidos. Las respuestas equivocadas indican que la regla necesita ser modificada, lo cual no es ningún problema.

A pesar de esto, veremos que RITA trabaja sorprendentemente bien en la mayor parte de las situaciones. Los límites que determinan cómo decide RITA cuáles son las conclusiones "más probables" y las "siguientes más probables" los calculé sencillamente mediante el método de prueba y error, sometiendo al sistema a una extensa variedad de ejemplos, cambiando los valores a medida que avanzaba.

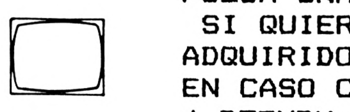
¿Es un gato o un perro?

Esta es una de las preguntas que han atormentado a los sabios desde los comienzos de la historia escrita.

Ahora, con la ayuda de FUZZY RITA, podremos determinarlo de una vez por todas.

Siguiendo a RITA en acción a través de este ejemplo artificial, nos resultará fácil comprender cómo trabaja y cómo puede utilizarse para nuestros propios problemas. El problema del GATO/PERRO es uno en el que la respuesta es "discreta"; *o* es gato *o* es perro. Los diagnósticos médicos frecuentemente pueden ser no discretos: *La evidencia indica una infección de las vías respiratorias superiores, o tal vez una forma benigna de sarampión.* FUZZY RITA puede funcionar con resultados discretos, y no necesita que se le diga a qué campo pertenece el problema. Las respuestas se dan siempre de la forma EL RESULTADO MAS PROBABLE ES... Si los datos suministrados a RITA, en conjunción con la regla base que ha creado, indican que hay uno o dos resultados casi igualmente probables, pueden dar una o dos conclusiones adicionales precedidas por las palabras EL SIGUIENTE MAS PROBABLE ES... Una vez que RITA ha aprendido la diferencia entre un gato y un perro, el usuario obtiene en general una única conclusión.

El programa comienza preguntándole si desea ver la base de conocimientos actualizada después de cada ejecución. Si es así, pulse cualquier tecla antes de RETURN.



```
PULSA UNA TECLA SEGUIDA DE < RETURN >
 SI QUIERES VER LOS CONOCIMIENTOS
ADQUIRIDOS DESPUES DE CADA RONDA.
EN CASO CONTRARIO PULSA UNICAMENTE
< RETURN > .
```

A continuación se introducen las "opciones de salida" (esto es, las posibles conclusiones que el sistema puede alcanzar), pulsando RETURN cuando haya

terminado de introducirlas. FUZZY RITA puede trabajar, tal y como se ha dicho, hasta con 50 salidas distintas:

```
---------------------------------------
INTRODUCE LA RESPUESTA NUMERO  1
 PULSA < RETURN > PARA TERMINAR
? PERRO
---------------------------------------
INTRODUCE LA RESPUESTA NUMERO  2
 PULSA < RETURN > PARA TERMINAR
? GATO
---------------------------------------
INTRODUCE LA RESPUESTA NUMERO  3
 PULSA < RETURN > PARA TERMINAR
?
```

Una vez que haya terminado de introducir las opciones, RITA las escribe en la pantalla, y le pide que introduzca las "preguntas" que el sistema formulará posteriormente. RITA puede trabajar con un máximo de 50 preguntas.

```
PERRO
GATO
---------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  1
 PULSA < RETURN > PARA TERMINAR
? COME RATONES Y MAULLA
---------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  2
 PULSA < RETURN > PARA TERMINAR
? ANIMAL
---------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  3
 PULSA < RETURN > PARA TERMINAR
? ASUSTA A LOS LADRONES
---------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  4
 PULSA < RETURN > PARA TERMINAR
```

Pulse otra vez RETURN para indicar que ha terminado de introducir las preguntas. Hecho esto, RITA le dará los objetos sobre los que puede decidir:

```
?
ESTOS SON LOS TEMAS ENTRE LOS QUE
PUEDO DISTINGUIR :
```

```
> PERRO
> GATO
------------------------------------
PIENSA UNO, DESPUES PULSA < RETURN >
```

Entonces se presentan las preguntas una a una, y se le pide al usuario que responda a ellas mediante un número comprendido entre 1 (lo que se pregunta es 100 por 100 cierto) y 0 (lo que se pregunta es 100 por 100 falso). Este le faculta a RITA para trabajar con datos discretos (del tipo GATO/PERRO, NEGRO/BLANCO) y con datos que cubren un rango de valores (como el peso, o la presión sanguínea de un paciente).

En el caso de datos que varían ampliamente (como el peso), es fácil concebir un sistema que nos permita representar el menor de ellos como el 0 y el mayor como el 1. Esto lo veremos después en un ejemplo sobre previsión climatológica, donde simplemente se dividen las temperaturas máximas y mínimas por 10, lo que implica que la gran mayoría de ellas se pueden introducir simplemente mediante un número tal como 0.6 (para 6 grados). Cualquier resultado superior a 1 se deja simplemente como 1. Las horas de sol se tratan de idéntica manera, siendo introducidas 7 horas de sol como 0.7, etc. Realmente no importa lo que sean los números, mientras se cree una regla que haga que el rango quede entre 1 y 0 y nos atengamos a ella al introducir tales números. Podemos tener diferentes reglas para distintas categorías de datos dentro de un mismo programa; a RITA no le afecta.

Al responder a las preguntas, podemos hacer también que el número que introducimos represente conceptos tales como "muy", "siempre" y "sin importancia". Por ejemplo, en preguntas como, ¿COMO ESTA DE ROJA LA CARA DEL PACIENTE? (0.9 significaría "muy", y 0.1 indicaría "poco"); ¿TIENE PROBLEMAS PARA ARRANCAR SU COCHE? (introduciríamos, por ejemplo, 0.9 para "siempre", y 0.4 para "algunas veces"); y ¿QUE IMPORTANCIA TIENE EL TIPO DE INTERES PARA ESTE SOLICITANTE DEL PRESTAMO? (1 para "es el aspecto más importante", pasando por 0.7 para "bastante importante", y 0.1 para "sin importancia").

Los números introducidos no tienen por qué ser exactos. Está claro que es difícil conseguir una equivalencia numérica consistente para conceptos tales como "casi nunca". Con RITA, basta casi siempre con aproximaciones, con tal que seamos razonablemente consistentes con una categoría particular de datos.

Como podemos ver, esto significa que RITA es capaz de habérselas con casi cualquier problema que le echemos, tanto si los datos son del tipo SI/NO, en una escala controlada (como la temperatura), o si lo son del tipo que admite interpretaciones. Nuestro ejemplo GATO/PERRO es casi totalmente, al menos para nuestros propósitos, una situación SI/NO.

Primero enseñamos a RITA algunas cosas sobre perros:

```
------------------------------------
PULSA UNO DE ESTOS NUMEROS :
1 (CIERTO) 0 (FALSO) $ (FIN)
```

```
COME RATONES Y MAULLA ?0
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

ANIMAL ?1
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

ASUSTA A LOS LADRONES ?1

EL RESULTADO MAS PROBABLE ES PERRO

     ES CORRECTO EL RESULTADO MAS
     PROBABLE  ?        ( S , N )
```

Por suerte, el sistema indica PERRO, la respuesta correcta (en realidad, no ha sido suerte, sino que sencillamente, cuando no tiene ninguna información sobre la que basar su decisión, elige la primera opción de salida).

RITA actualiza las reglas y nos informa de sus hallazgos:

```
----------------------------------------
PERRO

COME RATONES Y MAULLA 0
ANIMAL 2
ASUSTA A LOS LADRONES 4

----------------------------------------
GATO

COME RATONES Y MAULLA 0
ANIMAL 0
ASUSTA A LOS LADRONES 0

 PULSA < RETURN > PARA CONTINUAR EL
ADIESTRAMIENTO .
O CUALQUIER TECLA SEGUIDA DE <RETURN>
 PARA HACER USO DE RITA
```

Ha asignado un valor de 0 a la pregunta COME ARROZ Y MAULLA, 2 a ANIMAL y 4 a LADRA FEROZMENTE A LOS LADRONES. Esto no significa que por alguna razón piense que LADRA FEROZMENTE es más

importante que la categoría ANIMAL, ya que no dispone de información sobre la que tomar tal decisión. RITA multiplica el valor que se da a cada respuesta por un número que se duplica con cada pregunta; así, multiplica la respuesta a la primera pregunta por 1, la segunda por 2, la tercera por 4, la cuarta por 8... y así sucesivamente. Como vemos, hasta aquí, la regla base para GATO está todavía completamente llena de ceros. Esto es debido a que RITA no se ha encontrado todavía con un gato, y no sabe nada de ellos.

Presentemos un gato al programa, y veamos lo que hace con él:

```
ESTOS SON LOS TEMAS ENTRE LOS QUE
PUEDO DISTINGUIR :

 >  PERRO
 >  GATO
------------------------------------
PIENSA UNO, DESPUES PULSA < RETURN >
------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

COME RATONES Y MAULLA ?1
------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

ANIMAL ?1
------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

ASUSTA A LOS LADRONES ?O

EL RESULTADO MAS PROBABLE ES PERRO

    ES CORRECTO EL RESULTADO MAS
    PROBABLE  ?        ( S , N )
```

El sistema sugiere PERRO, ya que es lo que mejor encaja con los datos que él tiene (la pregunta ANIMAL obtuvo la misma respuesta). Una vez que se le dice que PERRO es incorrecta, RITA hace comprobaciones para ver cuántas salidas alternativas tiene. Si solamente tiene dos, sabe que la respuesta que no dio es la correcta para los datos presentados durante la ejecución, y ajusta sus reglas conforme a ello:

```
------------------------------------
PERRO

COME RATONES Y MAULLA 0
ANIMAL 2
ASUSTA A LOS LADRONES 4

------------------------------------
GATO

COME RATONES Y MAULLA 1
ANIMAL 2
ASUSTA A LOS LADRONES 0

 PULSA < RETURN > PARA CONTINUAR EL
ADIESTRAMIENTO .
O CUALQUIER TECLA SEGUIDA DE <RETURN>
 PARA HACER USO DE RITA
```

Podemos ver cómo RITA ha creado ahora la regla según la cual una respuesta afirmativa a COME RATONES y MAULLA y ANIMAL equivale a GATO, mientras que respuestas afirmativas a ANIMAL y a ASUSTA A LOS LADRONES equivalen a PERRO. A partir de ahora, en este caso, el más simple posible (dos salidas, preguntas S/N), RITA no volverá a fallar. Ha aprendido por sí misma a distinguir entre dos animales.

No tan sencillo

La situación no es tan fácil cuando hay más de dos salidas, pudiendo RITA necesitar varios funcionamientos de prueba antes de que esté segura de saber lo que ocurre. Incluso entonces preguntará si el resultado más probable que ha dado es correcto (y si le decimos que no, preguntará por el segundo más probable) y modificará sus reglas ligeramente si comete un error. No obstante, RITA dispone de un recurso extra que entra en juego cuando hay más de dos salidas. Puede decidir que determinadas preguntas son irrelevantes y no le ayudan a alcanzar sus conclusiones. (En el ejemplo GATO/PERRO, la pregunta ANIMAL era irrelevante; no daba a RITA ninguna información adicional para decidir qué clase de criatura era.)

Con la ayuda de un libro de texto elemental de química (White, 1979), decidí enseñar a RITA a distinguir entre magnesio, hierro y plomo. Esto, de hecho, no es difícil incluso para inexpertos como yo, pero es muy distinto de lo que tuve que hacer para indicarle la diferencia entre un gato y un perro, tema en el que soy un experto al 100 por 100.

Comenzaremos dando a RITA las opciones de salida:

```
PULSA UNA TECLA SEGUIDA DE < RETURN >
 SI QUIERES VER LOS CONOCIMIENTOS
ADQUIRIDOS DESPUES DE CADA RONDA.
EN CASO CONTRARIO PULSA UNICAMENTE
 < RETURN > .

------------------------------------
INTRODUCE LA RESPUESTA NUMERO  1
 PULSA < RETURN > PARA TERMINAR
? MAGNESIO
------------------------------------
INTRODUCE LA RESPUESTA NUMERO  2
 PULSA < RETURN > PARA TERMINAR
? HIERRO
------------------------------------
INTRODUCE LA RESPUESTA NUMERO  3
 PULSA < RETURN > PARA TERMINAR
? PLOMO
------------------------------------
INTRODUCE LA RESPUESTA NUMERO  4
 PULSA < RETURN > PARA TERMINAR
?
```

A continuación introducimos las preguntas discriminatorias:

```
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  1
 PULSA < RETURN > PARA TERMINAR
? SU DENSIDAD ES MENOR QUE 8 GM/CM^3
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  2
 PULSA < RETURN > PARA TERMINAR
? ES UN METAL
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  3
 PULSA < RETURN > PARA TERMINAR
? ES VENENOSO
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  4
 PULSA < RETURN > PARA TERMINAR
? CONDUCE LA ELECTRICIDAD
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  5
 PULSA < RETURN > PARA TERMINAR
? BRILLA CUANDO SE PULE
```

```
----------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  6
 PULSA < RETURN > PARA TERMINAR
? SE DESLUSTRA FACILMENTE
----------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  7
 PULSA < RETURN > PARA TERMINAR
?
```

RITA presenta un informe de la situación inicial y el entrenamiento comienza con el plomo como nuestro metal escogido:

```
ESTOS SON LOS TEMAS ENTRE LOS QUE
PUEDO DISTINGUIR :

 >  MAGNESIO
 >  HIERRO
 >  PLOMO
----------------------------------------
PIENSA UNO, DESPUES PULSA < RETURN >
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

SU DENSIDAD ES MENOR QUE 8 GM/CM^3 ?
0
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

ES UN METAL ? 1
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

ES VENENOSO ? 1
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

CONDUCE LA ELECTRICIDAD ? 1
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

BRILLA CUANDO SE PULE ? 0
```

```
-----------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

SE DESLUSTRA FACILMENTE ? 1

EL RESULTADO MAS PROBABLE ES MAGNESIO
EL SIGUIENTE MAS PROBABLE ES PLOMO


    ES CORRECTO EL RESULTADO MAS
    PROBABLE  ?        ( S , N ) ? N
ES CORRECTA MI SEGUNDA ELECCION  ?
          ( S , N ) ? S
```

Continuamos entrenándola unas cuantas veces más hasta que la base de conocimientos queda así:

```
-----------------------------------
MAGNESIO

SU DENSIDAD ES MENOR QUE 8 GM/CM^3 O
ES UN METAL O
ES VENENOSO O
CONDUCE LA ELECTRICIDAD O
BRILLA CUANDO SE PULE O
SE DESLUSTRA FACILMENTE O

-----------------------------------
HIERRO

SU DENSIDAD ES MENOR QUE 8 GM/CM^3 O
ES UN METAL O
ES VENENOSO O
CONDUCE LA ELECTRICIDAD O
BRILLA CUANDO SE PULE O
SE DESLUSTRA FACILMENTE O

-----------------------------------
PLOMO

SU DENSIDAD ES MENOR QUE 8 GM/CM^3 O
ES UN METAL 2
```

```
ES VENENOSO 4
CONDUCE LA ELECTRICIDAD 8
BRILLA CUANDO SE PULE 0
SE DESLUSTRA FACILMENTE 32
```

En esta ocasión, antes de pulsar la tecla RETURN pulsamos cualquier otra para indicar que el entrenamiento ha finalizado y que ya es hora de que RITA empiece a trabajar:

```
 PULSA < RETURN > PARA CONTINUAR EL
 ADIESTRAMIENTO .
O CUALQUIER TECLA SEGUIDA DE <RETURN>
 PARA HACER USO DE RITA
```

Un resultado práctico del cambio del modo de entrenamiento al de trabajo, aparte de un ligero cambio en la "conversación", es que ahora RITA decide qué preguntas hará, en vez de preguntarlas todas automáticamente:

```
ESTOS SON LOS TEMAS ENTRE LOS QUE
PUEDO DISTINGUIR :

 >  MAGNESIO
 >  HIERRO
 >  PLOMO
------------------------------------
ESTOY PREPARADA PARA DEDUCIR CUAL
TIENES EN MENTE.
------------------------------------
------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) 0 (FALSO) $ (FIN)

ES UN METAL ? 1
------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) 0 (FALSO) $ (FIN)

ES VENENOSO ? 0
------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) 0 (FALSO) $ (FIN)

CONDUCE LA ELECTRICIDAD ? 0
------------------------------------
```

```
 ---------------------------------------
  PULSA UNO DE ESTOS NUMEROS :
  1 (CIERTO) O (FALSO) $ (FIN)

 SE DESLUSTRA FACILMENTE ? 1

 EL RESULTADO MAS PROBABLE ES MAGNESIO

     ES CORRECTO EL RESULTADO MAS
     PROBABLE  ?        ( S , N ) ? S
```

A partir de ahora, solamente será actualizada la base de conocimientos si decimos a RITA que una de sus conclusiones está equivocada.

```
 PULSA < RETURN > PARA CONTINUAR EL
 ADIESTRAMIENTO .
 O CUALQUIER TECLA SEGUIDA DE < RETURN >
  PARA HACER USO DE RITA
 ESTOS SON LOS TEMAS ENTRE LOS QUE
 PUEDO DISTINGUIR :

  >  MAGNESIO
  >  HIERRO
  >  PLOMO
 ---------------------------------------
 ESTOY PREPARADA PARA DEDUCIR CUAL
 TIENES EN MENTE.
 ---------------------------------------
 ---------------------------------------
  PULSA UNO DE ESTOS NUMEROS :
  1 (CIERTO) O (FALSO) $ (FIN)

 ES UN METAL ? O
 ---------------------------------------
  PULSA UNO DE ESTOS NUMEROS :
  1 (CIERTO) O (FALSO) $ (FIN)

 ES VENENOSO ? 1
 ---------------------------------------
  PULSA UNO DE ESTOS NUMEROS :
  1 (CIERTO) O (FALSO) $ (FIN)

 CONDUCE LA ELECTRICIDAD ? O
 ---------------------------------------
```

```
----------------------------------------
 PULSA UNO DE ESTOS NUMEROS :
 1 (CIERTO) O (FALSO) $ (FIN)

SE DESLUSTRA FACILMENTE ? 1

EL RESULTADO MAS PROBABLE ES MAGNESIO

EL SIGUIENTE MAS PROBABLE ES PLOMO

    ES CORRECTO EL RESULTADO MAS
    PROBABLE  ?        ( S , N ) ? N
ES CORRECTA MI SEGUNDA ELECCION  ?
          ( S , N ) ? N
----------------------------------------
 1 - MAGNESIO
 2 - HIERRO
 3 - PLOMO

 CUAL ES LA SOLUCION CORRECTA ? ?  2


----------------------------------------
```



7

Razonamiento difuso

Mientras que la codificación de certidumbres en un programa de ordenador es razonablemente sencilla (SI esto es verdad ENTONCES esto es siempre verdad), la expresión de grados de certeza no es tan fácil. La lógica difusa, término introducido por L. A. Zadeh (1979), trata de la deducción de conclusiones a partir de premisas que carecen de precisión.

Las conclusiones sacadas se expresan en términos de *posibilidades* en vez de *probabilidades* que se utilizan cuando el grado de certidumbre de una premisa puede expresarse con exactitud en forma matemática. La mayoría de los mecanismos de razonamiento (tales como SI A ES SIEMPRE B, Y C ES A, ENTONCES C ES B) son incapaces de enfrentarse con imprecisiones y posibilidades.

La imprecisión acompaña a la mayoría de las acciones humanas, desde la comprensión del habla hasta la decisión del movimiento a realizar en una jugada de ajedrez. Tal como ya hemos visto, los ordenadores no se las arreglan bien con la imprecisión. Un bit o es 1 o es 0 (nunca "posiblemente 1"). La codificación de mecanismos que funcionen con situaciones confusas requiere una técnica particular de programación.

En lenguajes como PROLOG y HASTE (véanse los capítulos 11 a 13), el grado de certeza se codifica directamente. Al utilizar tales lenguajes declarativos podemos utilizar términos como *en la mayoría de los casos*, *a menudo*, *usualmente*, *a veces*, *casi nunca*, etc. Lo que es más, podemos unir y comparar los términos, de forma que podemos decir cosas como *Si es muy posible que A esté presente, entonces B casi nunca está presente.*

En los lenguajes humanos (o sea, naturales), hay una serie de términos que utilizamos en nuestras vidas cotidianas que dan grados de posibilidad a términos descriptivos. Lo siguiente aclarará lo que pretendo decir. Piense en un adjetivo como *rojo*, *largo* o *pesado*. Cuando describimos un objeto como rojo, largo o pesado, a menudo precedemos al objetivo con una palabra o frase que de alguna manera lo modifica, dándole un significado más exacto. Tales palabras o frases como *muy, solamente, no especialmente* y *en absoluto*, dan al adjetivo un grado de probabilidad. En tales casos, podemos decir que la "variable lingüística" se compone de dos partes, un *término primario* (como, por ejemplo, "rojo") y un *modificador*.

Aunque es claramente imposible el dar una estimación probabilística exacta a cada utilización de los modificadores particulares (como, por ejemplo, 0.95 para *muy* y 0.2 para *no especialmente*), resulta posible, en la práctica, asignarles valores basados en nuestra comprensión del grado de "fuerza" que un particular modificador confiere a un adjetivo, y —lo que es más importante— en nuestro conocimiento del objeto o de lo que sea que se esté describiendo y en el significado de tal adjetivo.

De vuelta al mundo real

Los ejemplos GATO/PERRO Y MAGNESIO/HIERRO/PLOMO tratados con RITA se idearon para exigir únicamente respuestas SI/NO. Ahora ha llegado el momento de ver si el programa puede funcionar también con datos menos definidos, y en una situación en la que no es fácil (o es imposible) hallar cuáles son las reglas. Una situación de este tipo es la predicción del tiempo. Aunque podemos mirar al cielo por la mañana, y decir algo profundo como "Parece que va a llover" o "Esperemos que aclare más tarde", probablemente disponemos de poca información objetiva con la que predecir cómo va a ser el tiempo el resto del día, o de mañana.

Vamos a observar los modelos meteorológicos de dos ciudades muy distintas, Londres y Melbourne. Miraremos en primer lugar y con bastante brevedad el modelo de Londres, utilizando para ello cuatro variables (horas de sol, temperaturas máximas y mínimas, y milímetros de lluvia). A pesar de utilizar tan sólo estas pocas variables, descubriremos que RITA se las arregla para funcionar extraordinariamente bien. El examen de la situación de RITA en este campo nos facilitará la comprensión del funcionamiento del programa cuando examinemos sus segmentos más importantes.

Una vez que hayamos observado el ejemplo de Londres, y nos abramos camino por el programa, le suministraremos un conjunto de sucesos meteorológicos más precisos relativos a Melbourne, y veremos si su rendimiento mejora conforme la información que se le da es más exacta.

Londres

Le dije a RITA que quería que eligiera una de las tres predicciones para el tiempo de un día, en base a la información del día anterior. Las predicciones posibles eran:

- Mañana lloverá bajo 1 mm.
- Mañana lloverá sobre 4 mm.
- Mañana lloverá entre 1 y 4 mm.

Las preguntas discriminatorias eran:

- Temperatura mínima (dividida por 10, para que tuviera un valor comprendido entre cero y uno, con cualquier resultado superior a 1, reducido a 1).
- Temperatura máxima (tratada de la misma manera que la temperatura mínima).
- Lluvia (tratada igual que las dos magnitudes anteriores).
- Horas de sol (divididas por 10. El resultado, que en este ejercicio nunca supere la unidad, se redondea a su primera cifra decimal).

Introducidos los datos de una semana, en la que RITA se encontró con días que correspondían a las tres posibles salidas, la regla base quedaba así:

```
-------------------------------------

PRONOSTICO DE LLUVIA BAJO 1 MM

TEMPERATURA MAXIMA (C  /10) .4
TEMPERATURA MINIMA (C  /10) .2
LLUVIA (MM /10) .5
HORAS DE SOL ( /10) 3.8

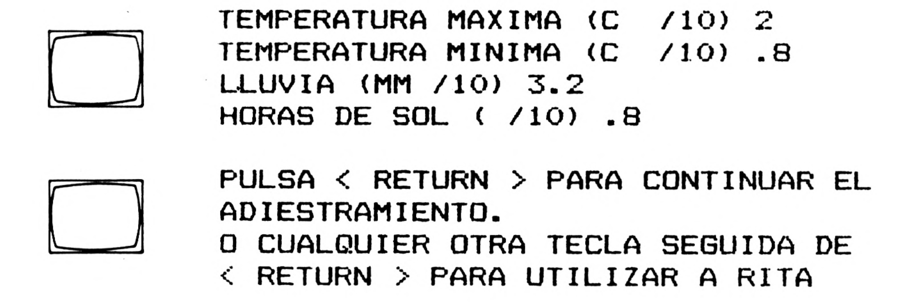
-------------------------------------

PRONOSTICO DE LLUVIA SOBRE 4 MM

TEMPERATURA MAXIMA (C  /10) 1.8
TEMPERATURA MINIMA (C  /10)  .3
LLUVIA (MM /10) 1.4
HORAS DE SOL (C /10) 35.4

-------------------------------------

PRONOSTICO DE LLUVIA ENTRE 1 - 4 MM
```



```
TEMPERATURA MAXIMA (C  /10) 2
TEMPERATURA MINIMA (C  /10) .8
LLUVIA (MM /10) 3.2
HORAS DE SOL ( /10) .8

PULSA < RETURN > PARA CONTINUAR EL
ADIESTRAMIENTO.
O CUALQUIER OTRA TECLA SEGUIDA DE
< RETURN > PARA UTILIZAR A RITA
```

Resulta muy interesante analizar esta regla base e intentar calcular qué normas ha concebido RITA, y ver la comparación de éstas con las típicas y toscas reglas empíricas que utilizamos al predecir el tiempo.

Para predecir una caída de lluvia inferior a un 1 mm el programa busca un valor de 0.4 para la temperatura mínima (casi lo mismo, 0.3, para lluvias superiores a 4 mm), mientras que un valor de 0.8 se señala como algo valioso cuando se intenta predecir lluvias entre 1 y 4 mm. Esto nos indica que RITA cree, después de tan sólo examinar los datos meteorológicos de una semana, que una alta (relativamente hablando) temperatura mínima conduce probablemente a una caída media de lluvia (o sea, entre 1 y 4 mm) al día siguiente, mientras que pequeños valores indican o bien lluvia escasa (o nula), o bien mucha lluvia (más de 4 mm). Esto, al menos para mí, es sorprendente.

No hay grandes variaciones en los valores asignados para la temperatura máxima (1.9, 1.8 y 2), por lo que quizá, para el mes en cuestión, esta magnitud no es demasiado importante. En contraste, la caída de lluvia durante el día en curso se considera como una variable importante. Una lluvia escasa (un valor de 0.5) sugiere un poco de lluvia al día siguiente, lo cual está de acuerdo con nuestra creencia sobre el tiempo, según la cual después de unos días secos vienen otros húmedos y así sucesivamente, y no que la meteorología de cada día sea algo independiente del día precedente. No obstante, en vez de afirmar que “fuertes lluvias hoy implican fuertes lluvias mañana”, RITA ha llegado a la conclusión de que un alto valor para la lluvia caída hoy (3.2) indica una lluvia media (1 a 4 mm) al día siguiente, y un valor moderado para lo llovido hoy (1.4) indica una fuerte lluvia al día siguiente. Después de reflexionar, esto parece razonable. Si la mayor parte de la lluvia cae hoy, es de suponer que quede menos por caer para mañana.

Finalmente nos fijaremos en las horas de sol y en lo que RITA hace con ellas. Las magnitudes muestran aquí una gran variación (aunque debemos tener presente que han sido multiplicadas por 16 antes de añadirse a la base de datos; lo importante es la *variación* dentro de una categoría particular y no el *valor* numérico *puro*). Para predecir un día con menos de 4 mm de lluvia, RITA busca mucho sol (en la regla base, hay un valor de 3.5), mientras que muy poco sol (0.8) indica más de 4 mm de lluvia y algo más (3.8) sugiere que el día siguiente será casi o completamente seco (menos de 1 mm).

Llegados a este punto, la regla base de RITA parece contener lo siguiente:

- RITA predice para mañana un día completamente húmedo (más de 4 mm de lluvia) si hoy la temperatura mínima es baja, llueve moderadamente y hay bastante sol.
- RITA predice para mañana un día seco (menos de 1 mm de lluvia) si hoy la temperatura mínima es baja (aunque ligeramente superior a la del caso anterior), llueve muy poco y —quizá sorprendentemente— hay muy poquito sol.
- RITA predice para mañana un día semihúmedo (1 a 4 mm de lluvia), cuando hoy la temperatura mínima es alta, llueve mucho (comparado con los dos casos anteriores) y la cifra de horas de sol es muy pequeña.

¿Qué tal lo hizo?

Aunque la interpretación de la regla base que RITA estableció en esta prueba es muy fácil (y extremadamente interesante), carece de valor, cualquiera que sea, si no faculta al sistema experto para predecir realmente el tiempo. Hice funcionar el programa durante diecinueve días más, anotando su primera predicción, y la segunda si la daba (dará una segunda predicción y quizá una tercera, si los datos no conducen directamente a una conclusión en particular), o si la primera era o no correcta. Si no lo era, comprobaba la segunda.

El resultado fue éste (después de diecinueve días):

- *Correctas (primera predicción solamente)*..................... 7 días
- *Segunda predicción correcta (si la primera fue errónea)*........ 7 días
- *Correctas (primera o segunda)*............................. 14 días
- *Totalmente erróneas (primera y segunda)*.................... 5 días

Es un buen resultado, si tomamos en consideración la respuesta "más probable" y la "siguiente más probable". Recordemos que no dimos a RITA ninguna regla. Simplemente introdujimos los datos, y dijimos si las predicciones hechas por el programa eran correctas o erróneas. RITA estableció las reglas, y a partir de muy pocos ejemplos se las arregló para hacer, en definitiva, los cálculos sobre el tiempo suficientes como para formular predicciones razonables. Resulta asombroso (al menos para mí) que, aunque el programa no "sabe" lo que hace, se las arregla para crear una regla que funciona, hasta cierto punto, en la "vida real".

¿Qué hubiera pasado si le hubiéramos dado más datos con los que seguir trabajando, o le hubiéramos pedido que sus predicciones sólo distinguieran entre días secos y días húmedos? ¿Habría funcionado RITA mejor? Intentaremos responder a estas preguntas, a su debido tiempo, con los datos meteorológicos de Melbourne.

RITA continuó aprendiendo durante un período completo de veintiséis días para el cual se introdujeron los datos. Decidí volver a probar otra vez con los datos del período inicial para ver si ahora su rendimiento había mejorado.

El resultado de sus hallazgos sobre el mismo período inicial de diecinueve días fue el siguiente:

- *Correctas (primera predicción solamente)*..... 9 días (aumenta en 2)
- *Segunda predicción correcta (si la primera fue errónea)*.................................... 8 días (aumenta en 1)
- *Correctas (primera o segunda)*............... 17 días (aumenta en 3)
- *Totalmente erróneas (primera y segunda)*.... 2 días (disminuye en 3)

Esta mejora me alegró muchísimo. Indicaba que, si añadíamos más y más ejemplos (en vez de utilizar los mismos una y otra vez, aunque esto parezca tener cierto grado de mérito), RITA continuaría aprendiendo. Una forma de dar a RITA cantidad de ejercicios (mejor que la tarea extremadamente pesada de introducirle por el teclado cantidad y cantidad de datos) podría ser la de poner los resultados registrados (esto es, las cifras de caída de lluvia y todo eso) en una matriz, junto con la respuesta correcta, y dejar simplemente que RITA elija días aleatoriamente durante una hora o más, refinando así incesantemente sus constantes de predicción.

Modificaciones

Tal como hemos visto, RITA elabora un número que se almacena para cada pregunta discriminatoria de cada salida posible. Si el sistema da una respuesta equivocada, modifica los números que ha almacenado para esa opción. Multiplica la cifra que guarda por cinco, le añade la nueva, divide la suma por seis y guarda el resultado. Esto asegura que un conjunto anómalo de información (como un gato que ladra ferozmente a los ladrones, o un día en el que ha caído 100 veces la lluvia de uno normal) no destruye totalmente los resultados más usuales. (No obstante a RITA le cuesta muchísimo deshacerse de la influencia de un conjunto de datos anómalos introducidos desde el principio.)

Probé con otras proporciones entre datos nuevos y datos viejos (como nueve veces el viejo, más el nuevo, dividido por diez; y el doble del viejo, más el nuevo, dividido por tres), pero esto implicaba que o bien el sistema aprendía con demasiada lentitud, o bien su base de datos fluctuaba violentamente en respuesta al último grupo de cifras con el que se enfrentaba. Lo mismo que el resto de RITA (y muchos otros sistemas expertos), el programa fue realizado según el método de prueba y error. Simplemente procedí según el principio "cámbialo si no va bien/déjalo si funciona".

Análisis de RITA

Vamos ahora a recorrer el programa RITA con cierto detalle: esta práctica que no he seguido en ninguna otra parte de este libro, la hago en esta ocasión porque RITA es el programa más útil de esta obra y es el que, con más

probabilidad, querrá usted adaptar para crear su propio sistema experto. Tal y como ya hemos dicho, RITA es un sistema de propósito general.

Anteriormente indiqué que tal vez se deseen modificar algunas de las cosas que hace el programa, cuando trabaja en campos específicos, para conseguir que RITA trabaje con mayor efectividad en este campo particular. Hágalo funcionar como si estuviera en su campo elegido, y entonces juzgue con él para conseguir tantas respuestas "correctas" como pueda, mientras la respuesta se conozca. De esta manera será mucho más probable que acierte cuando no se conozca la respuesta (resultado o salida) correcta. Después de todo, ésta es la única ocasión en la que los sistemas expertos se hacen realmente útiles, cuando se tienen los datos y se requiere que el programa tome alguna decisión en base a ellos. El programa comienza dimensionando una serie de matrices.

```
1370 REM ************************
1380 REM  INICIALIZACION
1390 CLS:KEYOFF
1400 REM REDUCE LAS MATRICES DE LA
     SIGUIENTE LINEA DE ACUERDO A
    TUS NECESIDADES O POSIBILIDADES
1410 DIM A$(50),B(50,50),C(50),D(50),
E$(50),F(50),E(50)
1420 X$=""
1430 PRINT "PULSA UNA TECLA SEGUIDA D
E < RETURN > SI QUIERES ";
1440 PRINT "VER LOS CONOCIMIENTOS"
1450 PRINT "ADQUIRIDOS DESPUES DE CAD
A RONDA."
1460 PRINT "EN CASO CONTRARIO PULSA U
NICAMENTE < RETURN > .":PRINT
1470 INPUT U$
1480 CLS
1490 RETURN
```

Los subíndices de estas matrices (los 50) son sobradamente superiores a lo que probablemente usted necesitará y ocupan bastante memoria (alrededor de 12K en mi equipo, un IBM PC). La matriz A$ contiene los nombres de las salidas, y E$ los de las preguntas discriminatorias; por tanto, sólo puede reducirlas fácilmente si conoce de antemano cuántas salidas habrá y cuántas preguntas formulará. Puede cambiar el resto de los subíndices de modo que igualen el número de preguntas discriminatorias. Otra alternativa podría ser modificar el programa de modo que le pidiera al principio el número de salidas y de preguntas y después dimensionara las matrices conforme a ello. Esta parte del programa consigue también un valor para U$ (vacío, o no) que determina si

el programa visualizará o no el contenido actualizado de la base de datos, después de cada ejecución.

A partir de aquí RITA sigue para aceptar los nombres de las posibles conclusiones que podrá alcanzar:

```
1160 REM   OPCIONES DE SALIDA
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
1200 PRINT "INTRODUCE LA RESPUESTA NU
MERO ";TT;"  PULSA < RETURN > PARA TE
RMINAR "
1210 INPUT A$(TT)
1220 IF A$(TT)="" OR TT=51 THEN TT=TT
-1:RETURN
1230 GOTO 1180
```

Y el de las preguntas discriminatorias que serán formuladas:

```
1250 REM  PREGUNTAS DISCRIMINATORIAS
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT " INTRODUCE LA PREGUNTA NU
MERO ";DQ;"  PULSA < RETURN > PARA TE
RMINAR "
1340 INPUT E$(DQ)
1350 IF E$(DQ)="" OR DQ=51 THEN DQ=DQ
-1:RETURN
1360 GOTO 1310
```

RITA hace preguntas al usuario en la parte del programa que comienza en la línea 140:

```
140 REM PREGUNTAS AL USUARIO
150 CLS
160 PRINT "ESTOS SON LOS TEMAS ENTRE
LOS QUE     PUEDO DISTINGUIR :"
170 PRINT
180 FOR J=1 TO TT
190 PRINT " >   ";A$(J)
200 NEXT J
210 GOSUB 1500
220 IF X$="" THEN PRINT "PIENSA UNO,
DESPUES PULSA < RETURN >"
230 IF X$<>"" THEN PRINT "ESTOY PREPA
RADA PARA DEDUCIR CUAL     TIENES EN M
ENTE."
240 IF X$="" THEN INPUT J$
250 ADD=.5
260 FOR J=1 TO DQ
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X$<>"" AND TT>2 THEN 390:REM
 REVISION POR SI LA PREGUNTA SE PUEDE
 PASAR POR ALTO
300 PRINT " PULSA UNO DE ESTOS NUMERO
S :"
310 PRINT " 1 (CIERTO) 0 (FALSO) $ (F
IN)
320 PRINT:PRINT E$(J);
330 INPUT H$:IF H$="$" THEN PRINT:PRI
NT " HASTA PRONTO ":PRINT:END
340 C(J)=VAL(H$)
350 C(J)=ADD*C(J)
360 NEXT J
370 RETURN
```

En la línea 330 se pide la introducción de una variable alfanumérica después de que en la línea precedente se haya visualizado la pregunta. Si la variable es un signo dólar, el programa finaliza (después de un educado GRACIAS) en la misma línea. En otro caso, el VAL de la variable se carga en un elemento de la matriz C (línea 340) que a continuación se multiplica en la línea siguiente por la variable ADD. Antes de que el bucle J comience (el cual admite las entradas del usuario), ADD se hace igual a 0.5 y se añade a sí misma (esto es, dobla su valor) cada vez que se recorre el bucle (haciéndola valer 1, 2, 4, 8, 16, y así sucesivamente) antes de que C(J) sea multiplicado por ella. Una vez recorrido el bucle el control del programa vuelve a un ciclo de llamadas a subrutinas próximo al comienzo del mismo.

La línea 290 envía el control del programa a la subrutina de la línea 390, la cual hace comprobaciones para ver si la pregunta se puede omitir.

```
380 REM **************************
390 REM DECIDE SI LA PREGUNTA SE
        PUEDE  OMITIR
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUM
P=0
430 NEXT W
440 IF JUMP=0 THEN 300
450 C(J)=B(W,J)
460 GOTO 360
```

Esto lo hace comparando todos los valores almacenados en la matriz B para aquella pregunta. Si la diferencia entre ellos no es mayor que 0.7, RITA supone que la información que suministra esa pregunta puede ser ignorada con toda seguridad. Este 0.7 es una de las constantes con las que podría querer jugar.

Una vez que todas las preguntas se han contestado, RITA va a la rutina más importante, situada a partir de la línea 480, donde se toma la decisión y —si es preciso— se actualizan las reglas.

```
470 REM **************************
480 REM TOMA DE DECISION
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXT J
520 ADD=.5
530 FOR J=1 TO TT


540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM  JUEGA CON LOS VALORES DE LAS
 TRES LINEAS SIGUIENTES PARA LOGRAR
 UNA MAYOR EFICACIA
570 IF C(X)=B(J,X) THEN D(J)=D(J)+1
580 IF ABS(C(X)-B(J,X))<.6*ADD THEN E
(J)=E(J)+.4
590 IF ABS(C(X)-B(J,X))<1.2*ADD THEN
F(J)=F(J)+.2
```

```
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
690 REM **   PRESENTA EL RESULTADO **
700 PRINT
710 CF=0
720 PRINT "EL RESULTADO MAS PROBABLE
ES ";A$(A1)
730 IF A2<>A1 THEN PRINT "EL SIGUIENT
E MAS PROBABLE ES ";A$(A2):CF=1
740 IF A3<>A2 AND A3<>A1 THEN PRINT "
EL SIGUIENTE MAS PROBABLE ES ";A$(A3)
:CF=2
750 PRINT
760 PRINT "     ES CORRECTO EL RESULTA
DO MAS          PROBABLE  ?        ( S
, N )";
770 INPUT F$
780 IF F$<>"S" AND F$<>"N" THEN 770
790 IF F$="S" AND X$<>"" THEN RETURN
800 IF F$="S" THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 9
80
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CF=0 THEN 890
840 PRINT "ES CORRECTA MI SEGUNDA ELE
CCION  ?             ( S , N )";
850 INPUT F$
860 IF F$="N" THEN 890
870 IF CF=1 THEN A1=A2:GOTO 980
880 IF CF=2 THEN A1=A3:GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;"- ";A$(J)
920 NEXT J
930 PRINT
940 PRINT " CUAL ES LA SOLUCION CORRE
CTA ?";
```

```
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM  **  EDUCANDO A RITA  **
   ( ACTUALIZACION DE CONOCIMIENTOS )
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+
5.5*B(A1,J))/6
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(A1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$="" THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K);B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
```

El proceso comienza igualando a cero los elementos de las matrices D, E y F. La matriz D almacenará los resultados "más probables", la E los "siguientes más probables" y la F los "siguientes más probables" después de E. Seguidamente, la variable ADD (que fue utilizada, como recordará, en la rutina "consultas al usuario", para multiplicar la información introducida por él) se iguala a 0.5 de manera que se pueda utilizar en el bucle siguiente.

El control del flujo del programa lo toman ahora un par de bucles anidados. El bucle J va desde uno hasta TT (número de salidas) con el bucle X yendo desde uno hasta DQ (número de preguntas discriminatorias).

Las tres líneas siguientes son las más importantes del programa. Es donde RITA toma su decisión. La línea 570 busca una coincidencia exacta entre la respuesta introducida, C(X), y aquel elemento de la matriz B que relaciona la posible salida (el valor que tiene TT en ese punto) y la pregunta cuya C(X) es la respuesta (el valor que DQ tiene en ese punto). Si encuentra una coincidencia total, D(J) se incrementa en uno. Es decir, la posibilidad de que la salida J sea el valor más alto —y por tanto de ser la respuesta seleccionada por RITA— se incrementa en una unidad. Tal vez se encuentre con que sus sistemas funcionan mejor, si en este punto se añade, por ejemplo, 1.5 en vez de 1.

La línea 580 busca una gran similitud entre un valor de la base de datos y la respuesta introducida, y si encuentra una coincidencia dentro de un margen de 0.6, añade 0.4 a E(J). Observe que, cuando una coincidencia es exacta, se

añade un 1 a D(J), y si es aproximada, un 0.4 a E(J). La línea 590 busca coincidencias "no tan exactas", y si las encuentra añade un 0.1 a F(J).

Las líneas 620 y 630 igualan a cero las variables A1, A2, A3, F1, F2 y F3, antes de activar el bucle J que va desde las líneas 640 a 680. Conforme recorre este bucle, RITA iguala cada variable F al valor más alto que puede hallar (haciendo F1 igual al mayor D(J); F2 igual al mayor E(J) y F3 al mayor F(J)). Cada vez que cambia F1, F2 o F3, se modifican A1, A2 o A3 para igualar al "número" de aquel elemento que provocó el cambio (es decir, se iguala al valor de J en ese punto). Esto le proporciona a RITA un registro de los elementos hallados hasta aquí que tengan el mayor valor.

Una vez que RITA recorre los bucles, A1 se iguala a la salida que tiene más probabilidades de ser cierta (debido a que la mayoría de las coincidencias y/o casi coincidencias entre la regla base y las respuestas del usuario se han dado para esa salida).

Ahora es el momento de que RITA anuncie su conclusión. La línea asigna el valor cero a la variable denominada CF. Como la matriz A$ contiene los nombres de las salidas, A$(A1) es el elemento de ella que es el nombre de la salida más probable. La línea 720 anuncia esta conclusión. Si A2 no es igual a A1 (es decir, el "siguiente más probable" no es el mismo que "el más probable"), RITA da este resultado y pone CF igual a 1 (que se usará en seguida). Si A3 tiene un valor diferente al de A1 y A2, se da un "siguiente más probable" y CF se iguala a 2.

La línea 770 pregunta si la conclusión de RITA es correcta. Si la respuesta es sí (es decir, F$ es igual a "S") y X$ no es igual a " " (lo que sucede cuando RITA deja de funcionar en modo aprendizaje y lo hace en modo de trabajo), la regla base no se modifica. Si RITA está en modo de trabajo, y la respuesta es correcta, entonces la regla base no debería ser alterada. Si RITA está todavía en modo aprendizaje, la línea 800 traslada la acción a la rutina que empieza en la línea 980, y actualiza la regla base.

Dos salidas

Si solamente hay dos salidas, la variable TT será igual a 2. El ordenador alcanza la línea 870 después de habérsele introducido una "N" (que indica que su respuesta era errónea). Por tanto, como sólo hay dos respuestas, la otra debe ser la correcta. Si A1 es igual a 1, entonces RITA dio —incorrectamente— la salida 1 como respuesta correcta. La línea 810 cambia esto, de modo que A1 se iguale a la respuesta correcta (es decir, a 2) antes de que la acción se traslade a la línea 980 para actualizar la regla base. La línea 880 hace lo contrario, cambiando un incorrecto 2 por un 1.

Si hay más de dos salidas, entonces a RITA se le ponen las cosas un poco más difíciles. Ya no es evidente cuál de las restantes salidas es la correcta. La línea 830 comprueba la variable CF y si la encuentra igual a cero, sabe que RITA no ha indicado ningún resultado "siguiente más probable", por lo que

salta a la subrutina de las líneas 890 a 960 que pregunta al usuario qué respuesta era la correcta.

Si RITA ha asignado algún valor a A2 y A3 (los "siguientes más probables") diferentes de los asignados a A1 (el "más probable"), el programa pregunta si su "segunda elección" A$(A2), es la correcta. Si es así, el valor de CF indica qué respuesta se ha dado como segunda elección (si CF es igual a 1, es el valor de A2; si CF es igual a 2, es el valor de A3) por lo que RITA sabe qué respuesta es correcta, y salta a la subrutina de la línea 980 equipada con esa información, con el fin de actualizar la regla base.

La siguiente subrutina, líneas 900 a 960, la hemos visto ya. Escribe en la pantalla todas las salidas, y pide al usuario que indique cuál es la correcta.

El programa recorre el bucle J desde las líneas 980 a 1020. Al recorrer la porción relevante de la base de datos, B(A1,J), comprueba si es igual a cero. Lo será si hasta el momento no se ha registrado todavía ninguna información (como ocurrirá siempre al comienzo de la ejecución del programa). Si B(A1,J) es diferente de cero (línea 990) el programa multiplica el valor actual guardado allí, por cinco, le añade el valor obtenido ahora y divide el resultado por seis. Esto asegura que: *a*) toda la información de experiencias anteriores no quede barrida por esta respuesta; *b*) el impacto de la respuesta actual no sea ignorado; *c*) una respuesta atípica no modifique la regla base demasiado (de modo que un único e inusual gato que nade no destruya la capacidad del programa para reconocer como gatos a animales que poseen todas las demás características "gatunas", pero que no saben nadar como nuestro anómalo gato.

Si B(A1,J) es igual a cero (línea 1000), entonces este elemento se iguala a la única respuesta que ha encontrado el sistema hasta aquí en relación con esta pregunta y salida, por lo que se hace igual a A1. La línea 1010 redondea cifras decimales. (Sin ella, RITA mantendría valores con hasta seis cifras decimales, lo cual es absurdo, dada la naturaleza altamente subjetiva de algunos de los datos originales.)

Finalmente, en esta larga y más importante parte del programa, RITA comprueba la línea 1040 para ver si U$ es igual a " ". Si lo es, significa que el usuario indicó al comienzo del programa que no quería ver en pantalla el estado actual de la regla base. En este caso, regresa al bucle de control del comienzo del programa para subsiguientes series de entradas. Si el usuario indicó su deseo de ver la regla base actual (y esto es, a mi entender, la parte más interesante de todo el proceso), la siguiente sección la escribe, tal y como hemos visto ya en los ejemplos del funcionamiento de RITA.



8
Listado completo de FUZZY RITA

En el capítulo anterior se vieron algunas de las partes del programa RITA. No obstante, no se dio el listado completo. Este capítulo salvará tal omisión. En el próximo, probaremos RITA con otro conjunto de datos meteorológicos, y mostraremos un medio de tratar los datos de entrada de forma que queden ordenadamente dentro de la escala del 0 al 1.

Antes de ello, no obstante, damos el listado:

```
10 REM     FUZZY RITA
20 GOSUB 1380:REM INICIALIZACION
30 GOSUB 1160:REM OPCIONES DE SALIDA
40 GOSUB 1250:REM  PREGUNTAS
                 DISCRIMINATORIAS
50 GOSUB 140:REM PREGUNTAS AL USUARIO
60 GOSUB 480:REM TOMA DE DECISION Y
   ACTUALIZACION DE CONOCIMIENTOS
70 PRINT " PULSA < RETURN > PARA CONT
INUAR";
80 IF X$<>"" THEN INPUT I$:GOTO 50
90 PRINT " EL ADIESTRAMIENTO ."
100 PRINT "O CUALQUIER TECLA SEGUIDA
DE <RETURN> PARA HACER USO DE RITA";
```

```
110 INPUT X$:GOTO 50
120 END
130 REM **************************
140 REM PREGUNTAS AL USUARIO
150 CLS
160 PRINT "ESTOS SON LOS TEMAS ENTRE
LOS QUE    PUEDO DISTINGUIR :"
170 PRINT
180 FOR J=1 TO TT
190 PRINT " >  ";A$(J)
200 NEXT J
210 GOSUB 1500
220 IF X$="" THEN PRINT "PIENSA UNO,
DESPUES PULSA < RETURN >"
230 IF X$<>"" THEN PRINT "ESTOY PREPA
RADA PARA DEDUCIR CUAL    TIENES EN M
ENTE."
240 IF X$="" THEN INPUT J$
250 ADD=.5
260 FOR J=1 TO DQ
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X$<>"" AND TT>2 THEN 390:REM
 REVISION POR SI LA PREGUNTA SE PUEDE
 PASAR POR ALTO
300 PRINT " PULSA UNO DE ESTOS NUMERO
S :"
310 PRINT " 1 (CIERTO) 0 (FALSO) $ (F
IN)
320 PRINT:PRINT E$(J);
330 INPUT H$:IF H$="$" THEN PRINT:PRI
NT " HASTA PRONTO ":PRINT:END
340 C(J)=VAL(H$)
350 C(J)=ADD*C(J)
360 NEXT J
370 RETURN
380 REM **************************
390 REM DECIDE SI LA PREGUNTA SE
        PUEDE  OMITIR
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUM
P=0
430 NEXT W
440 IF JUMP=0 THEN 300
```

```
450 C(J)=B(W,J)
460 GOTO 360
470 REM ************************
480 REM TOMA DE DECISION
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXT J
520 ADD=.5
530 FOR J=1 TO TT
540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM  JUEGA CON LOS VALORES DE LAS
 TRES LINEAS SIGUIENTES PARA LOGRAR
 UNA MAYOR EFICACIA
570 IF C(X)=B(J,X) THEN D(J)=D(J)+1
580 IF ABS(C(X)-B(J,X))<.6*ADD THEN E
(J)=E(J)+.4
590 IF ABS(C(X)-B(J,X))<1.2*ADD THEN
F(J)=F(J)+.2
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
690 REM **   PRESENTA EL RESULTADO **
700 PRINT
710 CF=0
720 PRINT "EL RESULTADO MAS PROBABLE
ES ";A$(A1)
730 IF A2<>A1 THEN PRINT "EL SIGUIENT
E MAS PROBABLE ES ";A$(A2):CF=1
740 IF A3<>A2 AND A3<>A1 THEN PRINT "
EL SIGUIENTE MAS PROBABLE ES ";A$(A3)
:CF=2
750 PRINT
760 PRINT "     ES CORRECTO EL RESULTA
DO MAS          PROBABLE  ?        ( S
, N )";
770 INPUT F$
780 IF F$<>"S" AND F$<>"N" THEN 770
790 IF F$="S" AND X$<>"" THEN RETURN
```

```
800 IF F$="S" THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 9
80
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CF=0 THEN 890
840 PRINT "ES CORRECTA MI SEGUNDA ELE
CCION  ?                ( S , N )";
850 INPUT F$
860 IF F$="N" THEN 890
870 IF CF=1 THEN A1=A2:GOTO 980
880 IF CF=2 THEN A1=A3:GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;"- ";A$(J)
920 NEXT J
930 PRINT
940 PRINT " CUAL ES LA SOLUCION CORRE
CTA ?";
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM  **  EDUCANDO A RITA  **
  ( ACTUALIZACION DE CONOCIMIENTOS )
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+
5.5*B(A1,J))/6
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(A1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$="" THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K);B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
1150 REM ************************
1160 REM   OPCIONES DE SALIDA
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
```

```
1200 PRINT "INTRODUCE LA RESPUESTA NU
MERO ";TT;"  PULSA < RETURN > PARA TE
RMINAR "
1210 INPUT A$(TT)
1220 IF A$(TT)="" OR TT=51 THEN TT=TT
-1:RETURN
1230 GOTO 1180
1240 REM ************************
1250 REM  PREGUNTAS DISCRIMINATORIAS
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT " INTRODUCE LA PREGUNTA NU
MERO ";DQ;"  PULSA < RETURN > PARA TE
RMINAR "
1340 INPUT E$(DQ)
1350 IF E$(DQ)="" OR DQ=51 THEN DQ=DQ
-1:RETURN
1360 GOTO 1310
1370 REM ***********************
1380 REM  INICIALIZACION
1390 CLS:KEYOFF
1400 REM REDUCE LAS MATRICES DE LA
     SIGUIENTE LINEA DE ACUERDO A
    TUS NECESIDADES O POSIBILIDADES
1410 DIM A$(50),B(50,50),C(50),D(50),
E$(50),F(50),E(50)
1420 X$=""
1430 PRINT "PULSA UNA TECLA SEGUIDA D
E < RETURN > SI QUIERES ";
1440 PRINT "VER LOS CONOCIMIENTOS"
1450 PRINT "ADQUIRIDOS DESPUES DE CAD
A RONDA."
1460 PRINT "EN CASO CONTRARIO PULSA U
NICAMENTE < RETURN > .":PRINT
1470 INPUT U$
1480 CLS
1490 RETURN
1500 PRINT STRING$(36,"-")
1510 RETURN
```



9

El observatorio de meteorología

En este capítulo convertiremos a RITA en un experto predictor del tiempo que hace en diciembre en Melbourne (Australia). Explicaré con algún detalle los pasos que he seguido para crear este sistema experto. De esta forma se puede hacer una idea de cómo utilizar a RITA para crear sus propios sistemas expertos reales.

En primer lugar, los datos tienen que estar de forma que a RITA le resulten fáciles de asimilar. Dijimos que RITA espera que se le introduzcan valores comprendidos entre cero y uno, siendo cero *falso* y uno *verdadero*, y representando los valores intermedios diferentes grados de veracidad. No tenemos por qué utilizar siempre valores entre cero y uno. RITA es extremadamente tolerante. Sin embargo, resulta más simple establecer un nivel estándar y mantenerlo siempre que se desarrolle un sistema experto a partir de la estructura de RITA, que tener que hacer pruebas y calcular posteriormente qué escala se está utilizando.

La relación de datos que le damos a RITA en este ejercicio corresponde a las condiciones meteorológicas de Melbourne, en diciembre de 1984. Daremos al ordenador las lecturas barométricas diarias efectuadas a las 9 horas de la mañana, las temperaturas mínimas y máximas y la humedad relativa a las 15 horas. A partir de estos datos, RITA ha de decirnos si al día siguiente va a llover o no.

El observatorio de meteorología de la Commonwealth (que proporcionó los datos) señaló que durante el mes que estamos estudiando se registró el

mayor número de días de lluvia (14) desde 1976. Esto está bien, ya que significa que alrededor de la mitad de los días del mes fueron húmedos, lo contrario que en 1982, mes casi totalmente seco que no habría representado un desafío importante para RITA y hubiera probado muy poco.

A continuación detallamos la relación de datos para los primeros días.

Fecha	*Baróm.*	*Temp. mín.*	*Temp. máx.*	*HR (%)*	*Lluvia*
1	1011.6	11.0	25.5	31	0
2	1006.7	12.6	27.6	29	0
3	1012.4	10.8	16.5	52	2.6

Se puede observar que los números varían mucho en magnitud, con unas lecturas barométricas alrededor de 1000, temperaturas que oscilan entre 10 y 30, con un porcentaje de humedad relativa que varía presumiblemente de cero a cien y lluvias de cero a infinito. ¿Cómo convertimos estos números en adecuados valores pequeños que queden en nuestra escala desde cero a uno?

Es muy fácil, y su ordenador hará casi todo el trabajo para usted. Cargue y ejecute el programa siguiente, y se lo explicaré.

```
10 REM HACIENDO ESCALAS
20 DIM X(50),Z(50)
30 CLS:KEYOFF
40 INPUT " VALOR MAXIMO :";A
50 INPUT " VALOR MINIMO :";B
60 A=A+1E-03
70 B=B-1E-03
80 C=(A-B)/50
90 X(O)=B
100 FOR J=1 TO 50
110 X(J)=X(J-1)+C
120 Z(J)=J/50
130 PRINT Z(J),X(J)
140 NEXT J
150 DF=(X(2)-X(1))/2
160 CN=0
170 CN=CN+1
180 PRINT "INTRODUCE EL VALOR";CN
190 INPUT Q$
200 IF Q$="" THEN END
210 Q=VAL(Q$)
220 IF Q<B OR Q>A THEN 180
230 FOR J=O TO 50
```

```
240 IF ABS(Q-X(J))<DF THEN LPRINT CN;
"-"; Z(J)
250 NEXT J
260 GOTO 170
```

Ejecute el programa y siga las indicaciones. Tenemos una relación de datos que han de ser introducidos, tales como los barométricos del boletín meteorológico. El programa nos pide el VALOR MAS ALTO?, así que echamos un vistazo a los datos barométricos en busca del mayor número de la lista. En la mía es el 1018.1, así que lo introduzco en el ordenador. La siguiente petición es VALOR MAS BAJO?, y buscándolo damos con el 994.2, que también lo introduzco.

Ahora el ordenador nos pide le introduzcamos uno a uno los datos que necesitamos, le da el número de orden (la variable COUNT en la línea 180) de la lista para el caso de que nos perdamos. Introducimos el primer número (1011.6) de nuestra lista, y la línea 240 escribe (en este caso en la impresora. Basta con quitar la L del LPRINT para que salga en la pantalla) el valor 0.72, que es el equivalente en la escala de cero a uno, de 1011.6 en la muchísimo menos adecuada escala de 994.2 a 1018.1. Recorremos todos los datos del mes, introduciendo cada uno de los números y anotando (o haciendo que el ordenador lo haga por nosotros) los resultados, de forma que los podamos introducir a RITA a su debido tiempo.

Las temperaturas mínimas, las máximas y la humedad relativa siguen el mismo proceso. Ahora estamos a punto de comenzar el adiestramiento de RITA sobre las peculiaridades del clima australiano. Primero, no obstante, haremos una serie de pequeñas modificaciones en el programa. Como recordará, dije que el programa RITA, tal cual, era únicamente una estructura en bruto que podía (y debía) ser modificado para que diera así los mejores resultados en el campo específico que queramos.

Como los datos a introducir para este programa vienen con dos cifras decimales, parece absurdo redondearlo a una sola cifra decimal en la línea 1010, con lo cual perdemos, probablemente, información vital. Para solucionarlo modificamos la línea 1010 de la siguiente manera:

```
1010 B(A1,J)=INT(100*B(A1,J))/100
```

La línea 570, que busca coincidencias exactas entre los datos que estamos introduciendo y los de la base de datos, se modifica para que busque coincidencias aproximadas, en vez de exactas, tal como sigue:

```
570 IF ABS(C(X)-B(J,X))<.2*ADD THEN
D(J)=D(J)+1
```

Adiestramiento de RITA

Ya estamos preparados para probar a RITA. Comenzamos por decirle que MAÑANA LLUVIA y MAÑANA SECO son las opciones de salida, y decidimos las preguntas discriminatorias que nos serán formuladas:

```
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  1
 (PULSA <RETURN> PARA TERMINAR)
? BAROMETRO
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  2
 (PULSA <RETURN> PARA TERMINAR)
? TEMPERATURA MINIMA
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  3
 (PULSA <RETURN> PARA TERMINAR)
? TEMPERATURA MAXIMA
------------------------------------
 INTRODUCE LA PREGUNTA NUMERO  4
 (PULSA <RETURN> PARA TERMINAR)
? HUMEDAD RELATIVA
------------------------------------
```

Transcurridos cuatro días de darle datos y hacer correcciones, la regla base de RITA queda así:

```
------------------------------------

AMANECER LLUVIOSO

BAROMETRO  .6
TEMPERATURA MAXIMA  .7
TEMPERATURA MINIMA  2.46
HUMEDAD RELATIVA  2.15

------------------------------------

AMANECER DESPEJADO

BAROMETRO  .72
TEMPERATURA MAXIMA  .48
TEMPERATURA MINIMA  2.88
HUMEDAD RELATIVA  1.12
```

Lo más destacado de la regla base creada por RITA se refiere a la humedad relativa. Si es baja, entonces es probable que el día siguiente sea seco, lo cual parece una regla razonable. También parece que RITA piensa que temperaturas mínimas bajas y altas lecturas barométricas señalan también hacia un día seco.

Hacemos funcionar el programa durante cuatro días más de datos. Después de ello, la anterior regla base de RITA ha sido sustituida por la siguiente:

```
-----------------------------------
AMANECER LLUVIOSO

BAROMETRO  .6
TEMPERATURA MAXIMA  .7
TEMPERATURA MINIMA  2.46
HUMEDAD RELATIVA  2.15

-----------------------------------
AMANECER DESPEJADO

BAROMETRO  .84
TEMPERATURA MAXIMA  .55
TEMPERATURA MINIMA  2
HUMEDAD RELATIVA  2.36
```

A pesar de que RITA mantiene sus opiniones anteriores sobre lecturas barométricas y temperaturas mínimas, su forma de pensar sobre la humedad relativa ha cambiado completamente. Continué pacientemente introduciendo más datos. Al final de mes la regla base de RITA se ha consolidado de la forma siguiente:

```
-----------------------------------
AMANECER LLUVIOSO

BAROMETRO  .43
TEMPERATURA MAXIMA  1
TEMPERATURA MINIMA  2.23
HUMEDAD RELATIVA  3.74

-----------------------------------
AMANECER DESPEJADO

BAROMETRO  .75
TEMPERATURA MAXIMA  .95
TEMPERATURA MINIMA  2.86
HUMEDAD RELATIVA  2.15
```

Una lectura barométrica baja, una alta temperatura mínima y una baja máxima, junto con una alta humedad relativa sugieren MAÑANA LLUVIA, mientras que las condiciones contrarias señalan a MAÑANA SECO. Estas ideas no parecen irracionales, pero ¿qué resultado dan en la práctica?

En el mes en cuestión, ignorando el día primero (ya que en su caso la respuesta de RITA depende por completo del orden en el que se han introducido las opciones de salida), había veintinueve días que podíamos comprobar. RITA predijo la presencia o ausencia de lluvia correctamente en dieciocho de ellos, lo cual parece bastante aceptable. Esta impresión queda reforzada si se examinan los datos, que muestran cómo RITA predijo como seco un día en el que sólo cayeron 0.2 mm de lluvia, y que unos pocos días húmedos en medio de una racha de secos fueron calculados acertadamente.

Para ver si RITA continúa aprendiendo, volvemos a introducir otra vez los datos de todo el mes. Al final de esta segunda prueba, RITA ha desarrollado esta regla base:

```
------------------------------------

AMANECER LLUVIOSO

BAROMETRO  .43
TEMPERATURA MAXIMA  1.01
TEMPERATURA MINIMA  2.13
HUMEDAD RELATIVA  3.97

------------------------------------

AMANECER DESPEJADO

BAROMETRO  .75

TEMPERATURA MAXIMA  .98
TEMPERATURA MINIMA  2.58
HUMEDAD RELATIVA  2.22
```

Podemos ver que básicamente RITA ha reforzado su anterior posición, incrementando un poco más su alto valor para la humedad relativa cuando se trata de un día lluvioso. Los resultados del segundo mes son un perfeccionamiento sobre los de la primera prueba. Los mismos veintinueve días han producido diecinueve predicciones correctas, aunque todavía sigue prediciendo como seco un día en el que han caído 0.2 mm de lluvia. También predijo acertadamente que el día segundo del mes sería seco (hecho que en la primera vez no tiene medios para hacerlo), elevando el índice de aciertos de la segunda vez a 20 de cada 30.

A continuación aparecen los resultados de RITA, indicando los errores con una X.

DIA	TIEMPO	PRIMERA PREDICCION	SEGUNDA PREDICCION
2	SECO	HUMEDO	SECO
3	HUMEDO	SECO X	HUMEDO
4	HUMEDO	SECO X	HUMEDO
5	HUMEDO	HUMEDO	HUMEDO
6	SECO	SECO	HUMEDO X
7	SECO	HUMEDO X	SECO
8	SECO	SECO	SECO
9	SECO	SECO	SECO
10	SECO	SECO	SECO
11	HUMEDO	SECO X	SECO X
12	HUMEDO	HUMEDO	HUMEDO
13	SECO	HUMEDO X	HUMEDO
14	HUMEDO	HUMEDO	HUMEDO
15	HUMEDO	SECO	SECO X
16	SECO	SECO	SECO
17	HUMEDO	SECO	HUMEDO
18	HUMEDO	SECO X	SECO X
19	SECO	HUMEDO X	HUMEDO X
20	HUMEDO	SECO X	SECO X
21	HUMEDO	SECO X	SECO X
22	SECO	SECO	SECO
23	SECO	SECO	SECO
24	SECO	SECO	SECO
25	HUMEDO	HUMEDO	SECO X
26	HUMEDO	HUMEDO	HUMEDO
27	HUMEDO	SECO X	SECO X
28	SECO	SECO	SECO
29	SECO	SECO	SECO
30	SECO	SECO	SECO

Una tabla como ésta sería muy práctica a la hora de intentar crear un sistema experto real, ya que destaca dónde se han producido los errores. Por ejemplo, en ambas ocasiones las predicciones para los días 18 a 21 fueron erróneas. Sería muy interesante descubrir los valores en ese punto, para ver si es preciso un pequeño retoque de algunas partes del proceso de discriminación.

Tal vez quiera ahora el lector hacerse con unas estadísticas de su propia ciudad y ver qué tal funciona RITA con ellas.

Nuevas entradas

Si quiere, podemos modificar la parte del programa de respuestas del usuario de modo que, en lugar de introducir un número entre 0 y 1, se

introduzca simplemente una palabra escogida de un menú, que se traduce internamente en un número adecuado para el sistema.

Tal menú (junto con el número que RITA podría crear en base a la respuesta dada después de cada palabra) podría ser así:

```
SELECCIONA LA OPCION CORRECTA PARA
ESTA CUESTION :
A - SIEMPRE                       (1)
B - CASI SIEMPRE                 (.8)
C - LA MITAD DE LAS VECES        (.6)
D - DE VEZ EN CUANDO             (.4)
E - MUY RARAMENTE                (.2)
F - NUNCA                         (0)
```

Tal vez se encuentre con que esto hace a su RITA no solamente más fácil de utilizar, sino también más efectivo, ya que es más probable obtener del usuario respuestas algo más consistentes que las que se obtendrían si tuviera que hacer una estimación de la veracidad de la respuesta a una pregunta discriminatoria.



10 Lógica y programación

La consecución de una máquina que funcione lógicamente es un paso fundamental en el camino seguido para lograr el comportamiento de una máquina que pudiera llamarse genuinamente inteligente. Las tentativas de programar el comportamiento lógico en una máquina poseen una dilatada historia en el estudio de la lógica.

El famoso silogismo de Aristóteles...

TODOS LOS HOMBRES SON MORTALES

ARISTOTELES ES UN HOMBRE

LUEGO ARISTOTELES ES MORTAL

...introdujo un concepto lógico fundamental, "esta conclusión se deduce de esta/s premisa/s".

Desafortunadamente, los ordenadores no están automáticamente orientados según esta línea de pensamiento por el tipo de lenguajes de programación de uso más frecuente en el momento presente. Actualmente la mayoría de ellos, incluido el BASIC, son *imperativos*. Es decir, están construidos casi por completo a base de comandos que tienen que ser obedecidos por el ordenador (LET X=95: LET Y=2+X: PRINT Y). Un lenguaje imperativo no es el mejor lenguaje para escribir programas que imiten el pensamiento lógico.

Lenguajes declarativos

Para ello necesitamos empezar a programar en lenguajes declarativos. En ellos, los programas se construyen a base de definiciones que describen la relación entre los elementos que el ordenador está manipulando. Cuando se ejecuta un programa imperativo, el ordenador sigue un número de órdenes, tomando decisiones del tipo IF/THEN, y luego da los resultados de su proceso. Al ejecutar un programa declarativo, el ordenador utiliza las definiciones para responder satisfactoriamente a una pregunta que versa sobre la relación entre elementos introducidos. La salida de tal programa es el vínculo que descubre.

La mayoría de los lenguajes de ordenador que hoy se utilizan, tales como el BASIC y el FORTRAN, funcionan muy bien cuando la tarea a realizar es "lineal", cuando la aproximación al problema requiere de un policía (la unidad central de proceso) que dirija el "tráfico del pensamiento" según una trayectoria bien definida. Pero tales aproximaciones no son adecuadas para las exigencias de la inteligencia artificial y de los sistemas expertos, donde un grupo de elementos necesitan poder relacionarse entre sí, libre y simultáneamente.

Los trabajos que se están llevando a cabo en centros como el Japan Institute for New Generation Computer Technology y en el UK Alvey Programme se apartan de la línea recta Von Newman que ha seguido la resolución de problemas mediante ordenador, desde los años cuarenta. El ordenador de la quinta generación, en vez de ser un único procesador operando secuencialmente, parece que va a ser un grupo de procesadores trabajando en paralelo, cada uno de ellos ocupado en sus tareas independientes (pero relacionadas y acopladas). Cada una de estas tareas es algo así como una subrutina de un programa principal, excepto que en vez de ser llamadas una a una, y solamente en ocasiones concretas durante la ejecución del programa, todos los "subprocesadores" están, informan y reaccionan constantemente a las salidas de los otros procesadores.

El trabajo de los equipos Alvey y japonés se han concentrado, en parte, en la utilización de lenguajes de programación descriptivos tales como el LISP *(List Proccesing*, Procesador de Listas) y sus derivados, como el PROLOG *(Programming in LOGIC*, Programación Lógica) y el LOGO. En esta sección examinaremos el LISP y el PROLOG (junto con otros dos lenguajes algo más simples —EASLE y HASTE— que desarrollé como introducciones al uso de lenguajes descriptivos o declarativos) y para cuando lleguemos al final de ella dispondrá de versiones para cada lenguaje para probarlas en su propio ordenador.

LISP

El LISP se remonta al año 1956, cuando se celebró en el Dartmouth College el primer seminario sobre inteligencia artificial. Fue organizado por cuatro hombres, entre los que se incluía un joven profesor de matemáticas, John McCarthy. Los cuatro propusieron a la Rockefeller Foundation el proyecto de convocar una conferencia sobre la premisa de que cualquier característica de la

inteligencia podría ser descrita con la suficiente exactitud como para hacer posible que una máquina la simulara (McCorduck, 1979).

Una de las ponencias de la conferencia, presentada por Herbert Simon, versó sobre un lenguaje para procesar listas (no demasiado elegante) que él había desarrollado, llamado IPL *(Information Processing Language*, Lenguaje para el Procesado de la Información). Chris Bidmead, escribiendo en la revista *Practical Computing* de octubre de 1984 (pág. 129), señaló que el IPL y la conferencia fueron la semilla que con el tiempo dieron a luz al LISP. "Sus (los del IPL) seudocódigos de bajo nivel y su sintaxis tipo *assembler* le sugirieron (a McCarthy) la idea de un lenguaje para el procesamiento de listas..."

McCarthy utilizó las ideas de Simon (junto con las de algunos otros que trabajaban en este campo, entre los que se incluían Alan Newell, J. C. Shaw y Gelernter, de IBM) para desarrollar el LISP. En 1958 tenía acabada y en funcionamiento una versión (LISP 1) y a partir de ella desarrolló LISP 1.5, que es el precursor de la mayoría de los LISP que hoy se utilizan, incluyendo (por supuesto) el programa SSLISP del final de esta parte del libro.

El LISP parte de dos tipos de datos: átomos y listas. Las listas se forman con átomos y/o otras listas. El LISP no utiliza programas como tales; en su lugar, evalúa listas. Para él los datos y los programas son, pues, lo mismo. Un programa en LISP se emplea pidiéndole que explore su base de datos en busca de una lista (o átomo) que cumpla ciertas condiciones.

PROLOG

PROLOG, el descendiente más vigoroso del LISP, utiliza en gran medida la falta de distinción entre datos y programa. Hasta cierto punto, un programa en PROLOG se compone de una base de datos formada por listas que pueden ser consultadas.

Alain Colmerauer inventó el lenguaje en los primeros años de la década de los setenta, y fue inicialmente desarrollado en Marsella en 1972 por Colmerauer y Roussell, como un intérprete escrito en ALGOL-W. Al año siguiente se volvió a escribir en FORTRAN. La nueva versión era notablemente más eficiente y fue rápidamente difundida a través del mundo académico en Europa y Estados Unidos. En la década siguiente a la de su implementación, universidades e investigadores en inteligencia artificial desarrollaron gradualmente sus propias versiones del lenguaje. El DEC-10 PROLOG de la Universidad de Edimburgo, que fue la primera versión que incorporó un compilador, es generalmente considerada como la implementación estándar del lenguaje.

La popularidad del PROLOG se ha incrementado enormemente desde que los japoneses anunciaron que la utilización del lenguaje sería uno de los principales elementos en su proyecto de inteligencia artificial quinta generación.

Micro-PROLOG

Actualmente disponemos de muchas versiones de PROLOG. La primera de ellas, micro-PROLOG, fue escrita por Frank G. McCabe, en el Imperial College de Londres, en la Logic Programming Unit. En 1982 apareció en *Assembler* Z80 para sistemas CP/M 2.2. Actualmente se puede conseguir para muchos ordenadores, incluyendo el IBM PC con sistema operativo MS-DOS o CP/M-86. La versión micro-PROLOG tiene una sintaxis mucho más sencilla comparada con la de otras versiones tales como la DEC-10 de Edimburgo. No obstante, puede ser fácilmente ampliada por el usuario, y es lo suficientemente potente como para desarrollar con ella trabajos provechosos.

Micro-PROLOG incluye un programa frontal llamado SIMPLE, con el que se trabaja con más facilidad que con el formato de listado tipo LISP que el mismo programa utiliza. A su debido tiempo, le proporcionaré un programa que emula al SIMPLE en micro-PROLOG, de forma que pueda aprender en cierta medida el lenguaje, sin tener que gastarse su dinero para comprarlo. Entonces estará preparado para decidir si está o no lo suficientemente interesado en el PROLOG como para que esté justificada la adquisición de un compilador para el lenguaje.

Un programa PROLOG consta de una base de datos de hechos y reglas que podemos interrogar. Para "iniciarle" con suavidad en los lenguajes declarativos o descriptivos, he inventado un lenguaje primitivo. Bastará con que introduzca en su ordenador un programa relativamente corto para que pueda utilizar en él este lenguaje. Los conocimientos que adquiera con este lenguaje inicial —dado en el próximo capítulo— pueden ser aplicados en nuestra versión de SIMPLE que estudiaremos más adelante.



11 Pensando en HASTE

Mi lenguaje se llama HASTE (de *HArtnell's Simple declarative TonguE*, Lenguaje declarativo de HARTNELL). En HASTE se construye una base de datos introduciendo sentencias que contienen un asterisco que las divide de modo efectivo en sujetos y predicados. El ordenador acepta estas sentencias y a partir de ellas puede contestar preguntas y alcanzar conclusiones. Esto es fácil de comprender si observamos la siguiente muestra del funcionamiento de HASTE. En primer lugar, le daremos al ordenador una serie de hechos:

```
> JUAN*ES UN HOMBRE
> PEDRO*TEME AL LOBO
> MARIA*TEME AL LOBO
> PEDRO*ES UN HOMBRE
> MARIA*ES UNA MUJER
> PEDRO*TREPA ARBOLES
> MARIA*TREPA ARBOLES
> UN DURO*SON CINCO PESETAS
> UNA LIBRA*SON CIEN PENIQUES
> PEDRO*MIDE DOS METROS
> PEDRO*ES MAYOR DE EDAD
> PEDRO*ES UN EXPERTO EN ORDENADORES
> MARIA*ES UN EXPEXTO EN ORDENADORES
> MARIA*MIDE TRES METROS
```

Observe el lugar donde queda emplazado el asterisco, precediendo directamente al verbo y tomando el lugar del espacio que normalmente aparecería en esa posición.

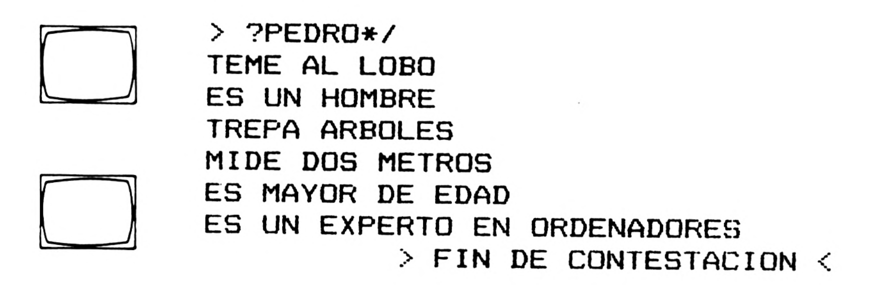
Para consultar la base de datos, se introduce un signo de interrogación una vez que el indicativo > aparece. Si quiere comprobar si un hecho particular está o no soportado por HASTE, se introduce el enunciado que queremos comprobar a continuación del signo de interrogación. El programa contesta con VERDAD o FALSO seguido de la línea FIN DE LA RESPUESTA para indicarle que la información que ha escrito es todo lo que puede darle como respuesta a esa pregunta.

Seguidamente haremos una comprobación para ver si HASTE ha aprendido algo sobre ciertos temas:

```
> ?PEDRO*ES UN HOMBRE
 CIERTO
                  > FIN DE CONTESTACION <
```

```
> ?PEDRO*TEME AL LOBO
 CIERTO
                  > FIN DE CONTESTACION <
```

Si deseamos conocer lo que HASTE sabe en relación a un tema particular, sustituimos en la línea de interrogación la información que queremos que nos suministre por un signo /. Seguidamente, HASTE nos revela todo lo que sabe en relación a PEDRO (dado que la pregunta, en definitiva, es "Dame todos los predicados que se refieren al sujeto PEDRO"):



```
> ?PEDRO*/
TEME AL LOBO
ES UN HOMBRE
TREPA ARBOLES
MIDE DOS METROS
ES MAYOR DE EDAD
ES UN EXPERTO EN ORDENADORES
                  > FIN DE CONTESTACION <
```

También podemos darle el predicado del enunciado, y HASTE nos dará todos los sujetos que tiene tal predicado:

```
> ?/*TEME AL LOBO
PEDRO
MARIA
                  > FIN DE CONTESTACION <
```

```
> ?UN DURO*/
SON CINCO PESETAS
            > FIN DE CONTESTACION <
```

Confío que practicando con este lenguaje limitado se haga una idea de cómo los lenguajes declarativos que pueden ser interrogados, pueden actuar como sistemas expertos de gran alcance.

Más útil que las formas anteriores de consultar es aquella según la cual el ordenador tiene que comprobar la veracidad de dos sentencias y suministrar información que satisfaga ambas condiciones. La siguiente pregunta, que utiliza un AND, pide a HASTE "¿Qué sujeto(s) tiene(n) miedo del lobo AND es un hombre?":

```
> ?/*TEME AL LOBO Y /*ES UN HOMBRE
PEDRO
            > FIN DE CONTESTACION <
```

O, "¿Qué sujeto(s) trepa(n) a los árboles AND es un experto en informática?":

```
> ?/*TEME AL LOBO Y /*ES UN EXPERTO E
N ORDENADORES
PEDRO
MARIA
            > FIN DE CONTESTACION <
```

Para algunas preguntas, no hay contestación:

```
> ?/*TREPA ARBOLES Y /*SON CIEN PENIQ
UES
            > FIN DE CONTESTACION <
```

Si HASTE fuera, por ejemplo, un sistema experto médico, le podríamos preguntar "¿Qué sujeto (enfermedad) produce granos y aparece en niños de menos de cuatro años?". La información que el sistema experto HASTE utilizaría, tendría que haber sido introducida en castellano corriente (aparte del asterisco). Esta es una de las ventajas reales de los lenguajes declarativos. Permiten utilizar el lenguaje natural (con algunas restricciones, por supuesto) para realizar las entradas y responde de una manera directa.

Si se desea descubrir todo lo que el sistema HASTE sabe en un momento dado, introducimos una / a cada lado del asterisco:

```
> ?/*/
JUAN*ES UN HOMBRE
PEDRO*TEME AL LOBO
```

```
MARIA*TEME AL LOBO
PEDRO*ES UN HOMBRE
MARIA*ES UNA MUJER
PEDRO*TREPA ARBOLES
MARIA*TREPA ARBOLES
UN DURO*SON CINCO PESETAS
UNA LIBRA*SON CIEN PENIQUES
PEDRO*MIDE DOS METROS
PEDRO*ES MAYOR DE EDAD
PEDRO*ES UN EXPERTO EN ORDENADORES
MARIA*ES UN EXPERTO EN ORDENADORES
MARIA*MIDE TRES METROS
```

Seguidamente aparecen los resultados de unas cuantas preguntas más:

```
> ?MARIA*/
TEME AL LOBO
ES UNA MUJER
TREPA ARBOLES
ES UN EXPERTO EN ORDENADORES
MIDE TRES METROS
          > FIN DE CONTESTACION <

> ?JUAN*/
ES UN HOMBRE
          > FIN DE CONTESTACION <

> ?PEDRO*/
TEME AL LOBO
ES UN HOMBRE
TREPA ARBOLES
MIDE DOS METROS
ES MAYOR DE EDAD
ES UN EXPERTO EN ORDENADORES
          > FIN DE CONTESTACION <
```

Antes de darle un listado de HASTE para que pueda experimentar su poder por sí mismo, vamos a dar un resumen de las normas de funcionamiento:

1. Todas las entradas tienen la forma de una sentencia, con un asterisco colocado entre el sujeto y el predicado.
2. Se interroga a la base de datos precediendo nuestra pregunta con un signo de interrogación.

3. Para comprobar si HASTE sabe algo sobre un hecho, introducimos el hecho precedido por un signo de interrogación. Contestará "VERDAD (lo sabe) o FALSO (no lo sabe)".
4. Una barra (/) sustituye en otras preguntas a la parte de la sentencia que deseamos que el programa nos conteste. Esto quiere decir que ?/*EL PADRE DE JULIO obtendrá como respuesta algo así como JUAN ES;?/*/ hará que se escriba toda la base de datos; y ?JUAN ES*/ obtendrá como respuesta algo así como EL PADRE DE JULIO.
5. La base de datos responde también a preguntas con AND dando respuestas para las cuales ambos enunciados son verdad, así ?JUAN/* AND EL PADRE/* nos devolverá toda la información que es verdad para ambos, JUAN y EL PADRE.

Como verá al analizar el listado, HASTE hace funcionar su magia manejando tan sólo los elementos de una pareja de matrices. Podemos almacenar en la base de datos hasta 255 hechos. A continuación viene el listado:

```
10 REM   HASTE
20 DIM A$(255),B$(255)
30 F=0:KEYOFF:CLS
40 REM *****************
50 FL=0
60 PRINT "> ";:LINE INPUT D$
70 IF D$="" THEN END
80 IF LEFT$(D$,1)="?" THEN 200
90 E=0
100 E=E+1
110 IF MID$(D$,E,1)="*" THEN 140
120 IF E< LEN(D$) THEN 100
130 PRINT " ENTRADA INVALIDA; NO OPER
ABLE":GOTO 50
140 IF FL=3 THEN RETURN
150 F=F+1:IF F=256 THEN END
160 A$(F)=LEFT$(D$,E-1)
170 B$(F)=MID$(D$,E+1)
180 GOTO 50
190 REM ****************
200 REM  INTERROGATORIO
210 FL=4:TR=0
220 IF RIGHT$(D$,1)="/" THEN FL=3
230 FOR J=1 TO LEN(D$)-5
240 IF MID$(D$,J,5)=" Y " THEN FL=5:T
R=J
250 NEXT J
```

```
260 IF FL=5 THEN 410
270 IF LEFT$(D$,3)="?/*" THEN FL=1
280 IF LEFT$(D$,4)="?/*/" THEN FL=2
290 IF FL=3 THEN GOSUB 90:F$=MID$(D$,
2,E-2)
300 IF FL=1 THEN F$=MID$(D$,4)
310 E=0
320 E=E+1
330 IF A$(E)="" AND FL=4 AND TR=0 THE
N PRINT " FALSO"
340 IF A$(E)="" THEN 520
350 IF FL=4 AND "?"+A$(E)+"*"+B$(E)=D
$ THEN PRINT " CIERTO":TR=1
360 IF FL=3 AND F$=A$(E) THEN PRINT B
$(E)
370 IF FL=2 THEN PRINT A$(E);"*";B$(E
)
380 IF FL=1 AND F$=B$(E) THEN PRINT A
$(E)
390 IF E<255 THEN 320
400 REM ****************
410 F$=MID$(D$,4,TR-4):G$=MID$(D$,TR+
7)
420 E=0
430 E=E+1
440 IF A$(E)="" THEN 520
450 IF B$(E)=F$ THEN 470
460 IF E<255 THEN 430
470 H=0
480 H=H+1
490 IF B$(H)="" THEN 460
500 IF B$(H)=G$ AND A$(E)=A$(H) THEN
PRINT A$(E)
510 IF H<255 THEN 480
520 PRINT TAB(5);" > FIN DE CONTESTAC
ION < "
530 GOTO 50
```



12

Una experiencia con PROLOG

Ahora que hemos adquirido algo de experiencia en la forma de trabajar con un lenguaje declarativo, podemos avanzar hasta el PROLOG. El programa que le voy a dar le permitirá ejecutar a su ordenador una versión restringida del programa frontal del PROLOG, SIMPLE. La he llamado PROLOG-A (que viene de PROLOG-*Almost*, es decir, PROLOG aproximado).

A pesar de que debería ser capaz de aprender una gran cantidad de cosas relativas al PROLOG con sólo leer esta parte del libro, y con experimentar con el programa, no se pretende que sea un verdadero manual del lenguaje. No obstante, si toma un libro sobre PROLOG o sobre micro-PROLOG, se dará cuenta de que puede utilizar el programa dado aquí, en conjunción con su libro, para aprender los fundamentos de manejo del frontal SIMPLE del lenguaje.

Antes de analizar el PROLOG-A, veremos a continuación una muestra del programa PROLOG funcionando sobre un compilador completo (lo que quede encerrado entre /*...*/ es comentario, lo mismo que una sentencia REM en un programa de BASIC).

```
/* una base de datos día/noche */

búho es-un nocturno
murciélago es-un nocturno
gato es-un diurno
perro es-un diurno
```

```
/* reglas */

X es-un durmiente-diurno si X es-nocturno
X es-un durmiente-nocturno si X es-diurno

X combate Y si X es-un durmiente-diurno
e Y es-un durmiente-nocturno
```

Como observará, hemos establecido unos *hechos* iniciales, utilizando la forma es-un. A continuación, le hemos dado al ordenador algunas *reglas* que los relacionan. Ahora podemos hacer preguntas al ordenador como éstas:

```
Es (búho es-un durmiente-diurno)
SI

cuál((XY): X combate Y)
respuesta es (gato búho)
respuesta es (gato murciélago)
respuesta es (perro búho)
respuesta es (perro murciélago)
```

Vemos aquí que el ordenador utiliza las reglas que se le han dado para responder a las preguntas. Esto es, fundamentalmente, lo que hace un sistema experto. Es fácil reconocer a PROLOG como un lenguaje que parece construido ex profeso para la realización de sistemas expertos.

Puede resultar incluso más útil el pedir a un programa PROLOG que nos explique cómo ha llegado a sus conclusiones:

```
    es (gato combate búho)
    SI

    por qué (gato combate búho)
    (gato combate búho) dado que
    (gato es-un durmiente-nocturno) dado que
        (gato es-un diurno) y
        (diurno es-un durmiente-nocturno)
    (búho es-un durmiente-diurno) dado que
        (búho es-un nocturno) y
        (nocturno es-un durmiente-diurno)

(durmiente-nocturno combate durmiente-diurno) conclusión
```

Como puede ver, PROLOG es bastante más complejo que HASTE, aunque en definitiva son miembros de la misma familia de lenguajes. Confío en que sus experiencias con HASTE le hayan facilitado la comprensión de las muestras de PROLOG que acabamos de observar.

Ahora podemos ver cómo funciona PROLOG-A.

El ejemplo de los marcianos

Comenzamos, al igual que hicimos con HASTE, introduciendo hechos en la base de datos. En PROLOG-A el indicativo es &.; el punto es un indicativo para el ordenador que está explorando el teclado, en espera de una entrada. Conseguimos que el programa añada hechos a su base de datos mediante un comando muy adecuado, METE. Mientras que en HASTE lo que introducíamos se componía de dos partes, un sujeto y un predicado (que incluía un verbo), en PROLOG-A el material introducido consta de tres partes:

1. *Un sujeto (tal como FELIPE).*
2. *Una relación (tal como ES-EL-PADRE-DE).*
3. *Otro sujeto o hecho (tal como ALFONSO).*

Tal como vemos, según el número 2 anterior, en cada sección las palabras se unen mediante guiones. Las secciones se separan entre sí mediante espacios. Tanto los espacios como los guiones son muy importantes.

Digámosle al programa unas cuantas cosas:

```
&.METE(ZAPA ES-UN MARCIANO)

&.METE(ERON ES-UN MARCIANO)

&.METE(MAIZ ES-UN VEGETAL)

&.METE(AGUA ES-UN LIQUIDO)

&.METE(LIQUIDO ES-UN BREBAJE)

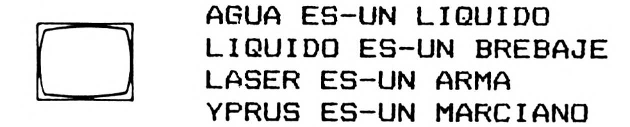
&.METE(LASER ES-UN ARMA)

&.METE(YPRUS ES-UN MARCIANO)
```

Podemos determinar lo que sabe mediante el comando LISTA TODO, que simplemente le dice a PROLOG que liste todos los hechos que contiene su base de datos:

```
&.LISTA TODO

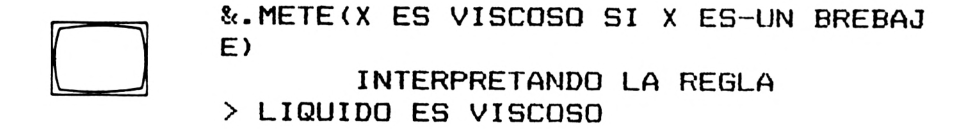
ZAPA ES-UN MARCIANO
ERON ES-UN MARCIANO
MAIZ ES-UN VEGETAL
```



```
AGUA ES-UN LIQUIDO
LIQUIDO ES-UN BREBAJE
LASER ES-UN ARMA
YPRUS ES-UN MARCIANO
```

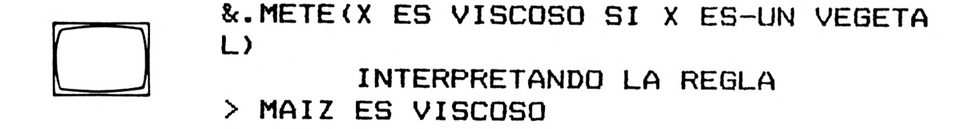
Ahora llegamos a la parte de PROLOG verdaderamente excitante, donde vemos una habilidad que HASTE no poseía. Es su capacidad para combinar por sí misma diferentes entradas para añadir nuevas informaciones a su base de datos. Hasta ahora, solamente le hemos dado *hechos*, que ha almacenado debidamente. Ahora le enseñaremos algunas *reglas* que relacionan algunos de ellos.

Aquí viene la primera regla:



```
&.METE(X ES VISCOSO SI X ES-UN BREBAJ
E)
        INTERPRETANDO LA REGLA
> LIQUIDO ES VISCOSO
```

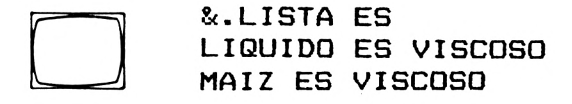
Esto le dice que un objeto (X es una variable en PROLOG) ES ALIMENTO si ese objeto ES-UNA BEBIDA. El programa nos dice que compila la regla, y escribe todo lo que ha añadido a su base de datos como resultado de aplicar esa regla a los hechos que conoce. Le damos una segunda regla:



```
&.METE(X ES VISCOSO SI X ES-UN VEGETA
L)
        INTERPRETANDO LA REGLA
> MAIZ ES VISCOSO
```

Como dije en el párrafo anterior, X es una variable en PROLOG. Micro-PROLOG utiliza variables, X, Y, Z, x, y, z, pero nosotros nos quedaremos sólo con X e Y. Las variables son como cajas vacías en las que podemos colocar los contenidos apropiados. *No podemos* utilizar un nombre de la base de datos como nombre de una variable (así, no podemos llamar X a uno de nuestros marcianos, o el programa incurriría en grandes confusiones).

Seguidamente le pedimos a PROLOG-A que nos diga todo lo que sabe que incluya la relación ES (que es, como advertirá, muy distinta de la relación ES-UN):



```
&.LISTA ES
LIQUIDO ES VISCOSO
MAIZ ES VISCOSO
```

Y así continuamos, dejando que PROLOG-A construya su bagaje de hechos relacionados con el universo en el que será un experto.

```
&.METE(X HABLA MARCIANO SI X ES-UN MA
RCIANO)
       INTERPRETANDO LA REGLA
> ZAPA HABLA MARCIANO
> ERON HABLA MARCIANO
> YPRUS HABLA MARCIANO
```

El PROLOG tiene también la posibilidad de compilar reglas que relacionan a más de una variable:

```
&.METE(X PROGRAMA-SU Z SI X HABLA MAR
CIANO Y Z ES-UN ARMA)
          INTERPRETANDO LA REGLA
> ZAPA PROGRAMA SU LASER
> ERON PROGRAMA SU LASER
> YPRUS PROGRAMA SU LASER
```

Aquí se le ha comunicado al programa una relación (PROGRAMA-SU) entre las variables X y Z, que existe si X HABLA MARCIANO y Z ES-UN ARMA.

Conforme avanzamos, adiestrando gradualmente a PROLOG-A para que actúe como un sistema experto en el dominio especificado de Marte, sus habitantes y estilo de vida, podemos comprobar lo que está aprendiendo. ¿Sabe, por ejemplo, que las pistolas laser son peligrosas?

```
&.ES(LASER ES PELIGROSO)
 NO
```

No lo sabe, así que decidimos enseñarle:

```
&.METE(X ES PELIGROSO SI X ES-UN ARMA
)
     INTERPRETANDO LA REGLA
> LASER ES PELIGROSO
```

Hacemos una comprobación para ver si ha aprendido la lección (utilizando un ES para indicarle que le estamos formulando una pregunta).

```
&.ES (LASER ES PELIGROSO)
 SI
```

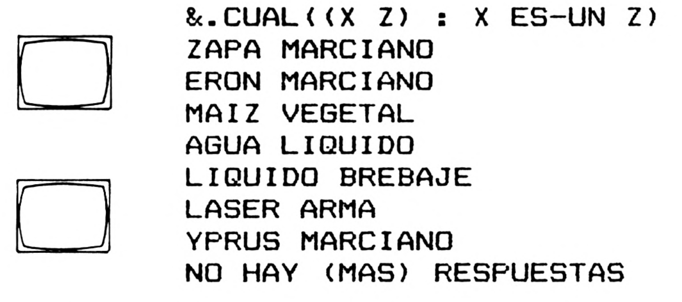
Podemos aprender mucho interrogando a la base de datos. Le podemos hacer preguntas en función de relaciones:

```
&.LISTA HABLA
ZAPA HABLA MARCIANO
ERON HABLA MARCIANO
YPRUS HABLA MARCIANO
```

O, como con HASTE, podemos utilizar variables en la pregunta, mediante el comando CUAL:

```
&.CUAL(X : X ES-UN MARCIANO)
ZAPA
ERON
YPRUS
NO HAY (MAS) RESPUESTAS
```

Aquí pedimos CUAL sujeto es tal que resulta verdad decir que tal sujeto ES-UN MARCIANO. Tenga en cuenta los espacios en la forma interrogativa. Son vitales para el funcionamiento satisfactorio de PROLOG-A. No es preciso que nos restrinjamos a una única variable:



```
&.CUAL((X Z) : X ES-UN Z)
ZAPA MARCIANO
ERON MARCIANO
MAIZ VEGETAL
AGUA LIQUIDO
LIQUIDO BREBAJE
LASER ARMA
YPRUS MARCIANO
NO HAY (MAS) RESPUESTAS
```

Esta pregunta dice, en efecto, CUAL dos cosas (X e Y) son tales que se relacionan mediante ES-UN. Observe que PROLOG-A escribe los valores de X e Y que encuentra sin escribir también la relación. Además, observe que (como en el ejemplo anterior) las respuestas de PROLOG-A finalizan con la frase NO HAY (MAS) RESPUESTAS para indicar que nos ha dado toda la información que puede sobre esa pregunta.

La última habilidad que PROLOG-A posee es la de ser capaz de compilar reglas en las que hay dos variables. Veremos que PROLOG-A, a pesar de su inteligencia, no puede trabajar con información que no posee:

```
&.METE(X BEBE Z SI X HABLA MARCIANO Y
 Z ES-UN BREBAJE)
> ZAPA BEBE LIQUIDO)
> ERON BEBE LIQUIDO)
> YPRUS BEBE LIQUIDO)
```

```
&.LISTA ES-UN

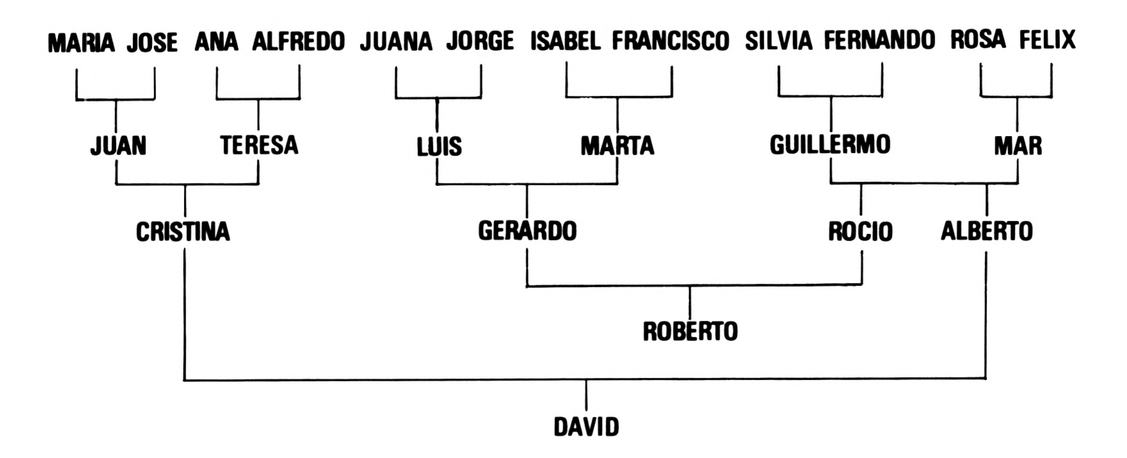
ZAPA ES-UN MARCIANO
ERON ES-UN MARCIANO
MAIZ ES-UN VEGETAL
AGUA ES-UN LIQUIDO
LIQUIDO ES-UN BREBAJE
LASER ES-UN ARMA
YPRUS ES-UN MARCIANO


&.METE(X SE-SIENTE-MAL-COMIENDO Z SI
X HABLA MARCIANO Y Z ES-UN VEGETAL)
> ZAPA SE-SIENTE-MAL-COMIENDO MAIZ
> ERON SE-SIENTE-MAL-COMIENDO MAIZ
> YPRUS SE-SIENTE-MAL-COMIENDO MAIZ
```

Familias felices

La demostración anterior era meramente instructiva y de entretenimiento, por lo que su dominio, Marte, era totalmente imaginario y sus relaciones estaban ideadas para mostrar el funcionamiento de PROLOG-A. Para demostrar que hay cosas más reales que la vida en Marte, observaremos una base de datos "verdadera", mucho más próxima a la Tierra.

Vamos a enseñarle a nuestro programa el siguiente árbol familiar:



Comenzamos con la primera generación:

```
&.METE(MARIA MADRE-DE JUAN)

&.METE(JOSE PADRE-DE JUAN)

&.METE(ANA MADRE-DE TERESA)

&.METE(ALFREDO PADRE-DE TERESA)

&.METE(JUANA MADRE-DE LUIS)

&.METE(JORGE PADRE-DE LUIS)

&.METE(SILVIA MADRE-DE GUILLERMO)

&.METE(FERNANDO PADRE-DE GUILLERMO)

&.METE(ISABEL MADRE-DE MARTA)

&.METE(FRANCISCO PADRE-DE MARTA)

&.METE(ROSA MADRE-DE MAR)

&.METE(FELIX PADRE-DE MAR)
```

Hacemos una comprobación para ver lo que en este momento sabe ya PROLOG-A:

```
&.LISTA TODO

MARIA MADRE-DE JUAN
JOSE PADRE-DE JUAN
ANA MADRE-DE TERESA
ALFREDO PADRE-DE TERESA
JUANA MADRE-DE LUIS
JORGE PADRE-DE LUIS
SILVIA MADRE-DE GUILLERMO
FERNANDO PADRE-DE GUILLERMO
ISABEL MADRE-DE MARTA
FRANCISCO PADRE-DE MARTA
ROSA MADRE-DE MAR
FELIX PADRE-DE MAR
```

Añadimos unas cuantas ramas más a la base de datos del árbol familiar:

```
&.METE(TERESA MADRE-DE CRISTINA)

&.METE(JUAN PADRE-DE CRISTINA)

&.METE(MARTA MADRE-DE GERARDO)

&.METE(LUIS PADRE-DE GERARDO)

&.METE(MAR MADRE-DE ROCIO)

&.METE(GUILLERMO PADRE-DE ROCIO

&METE(MAR MADRE-DE ALBERTO)

&.METE(GUILLERMO PADRE-DE ALBERTO)

&.METE(ROCIO MADRE-DE ROBERTO)

&.METE(GERARDO PADRE-DE ROBERTO)

&.METE(CRISTINA MADRE-DE DAVID)

&.METE(ALBERTO PADRE-DE DAVID)
```

En el dominio relativo a los marcianos, las relaciones eran del tipo ES-UN o BEBE. En esta segunda base de datos, tenemos PADRE-DE y MADRE-DE, lo que significa que podemos pedir, por ejemplo, todas las MADRE-DE que contiene:

```
&.LISTA MADRE-DE
MARIA MADRE-DE JUAN
ANA MADRE-DE TERESA
JUANA MADRE-DE LUIS
ISABEL MADRE-DE MARTA
SILVIA MADRE_DE GUILLERMO
ROSA MADRE-DE MAR
TERESA MADRE-DE CRISTINA
MARTA MADRE-DE GERARDO
MAR MADRE-DE ROCIO
MAR MADRE-DE ALBERTO
ROCIO MADRE-DE ROBERTO
CRISTINA MADRE-DE DAVID
```

Podemos comprobar también los PADRE-DE:

```
&.LISTA PADRE-DE
JOSE PADRE-DE JUAN
ALFREDO PADRE-DE TERESA
JORGE PADRE-DE LUIS
FRANCISCO PADRE-DE MARTA
FERNANDO PADRE-DE GUILLERMO
FELIX PADRE-DE MAR
JUAN PADRE-DE CRISTINA
LUIS PADRE-DE GERARDO
GUILLERMO PADRE-DE ROCIO
GUILLERMO PADRE-DE ALBERTO
GERARDO PADRE-DE ROBERTO
ALBERTO PADRE-DE DAVID
```

La pregunta más simple que podemos formular a PROLOG-A es aquella que comprueba si un hecho es o no verdad:

```
&.ES(CRISTINA MADRE-DE LUIS)
 NO

&.ES(LUIS PADRE-DE GERARDO)
 SI

&.ES(JUANA MADRE-DE MAR)
 NO
```

Tengamos en cuenta que, al igual que con los marcianos, PROLOG-A responde un SI o un NO. Nuestra última pregunta era si Juana era la madre de Mar. PROLOG-A nos dice que no. Por tanto, es razonable que pudiéramos preguntarle quién es la madre de Mar.

```
&.CUAL(X : X MADRE-DE MAR)
ROSA
NO HAY (MAS) RESPUESTAS
```

Esta pregunta, lo mismo que algunas que vimos en el ejemplo de Marte, está realmente preguntando "¿Cuál X es tal que X es la MADRE-DE MAR?". Los padres admiten el mismo tratamiento:

```
&.CUAL(X : X PADRE-DE JUAN)
JOSE
NO HAY (MAS) RESPUESTAS
```

Podemos ahora formular una pregunta que utilice dos variables, para hallar los nombres de todas las MADRES-DE en el universo del conocimiento de PROLOG-A:

```
&.CUAL((X Z) : X MADRE-DE Z)
MARIA JUAN
ANA TERESA
JUANA LUIS
ISABEL MARTA
SILVIA GUILLERMO
ROSA MAR
TERESA CRISTINA
MARTA GERARDO
MAR ROCIO
MAR ALBERTO
ROCIO ROBERTO
CRISTINA DAVID
NO HAY (MAS) RESPUESTAS
```

Hasta el momento hemos utilizado la relación MADRE-DE y PADRE-DE. Creo que es el momento de hablarle a PROLOG-A de parientes (padre o madre):

```
JUAN PARIENTE-DE CRISTINA
LUIS PARIENTE-DE GERARDO
GUILLERMO PARIENTE-DE ROCIO
GERARDO PARIENTE-DE ROBERTO
GUILLERMO PARIENTE-DE ALBERTO
ALBERTO PARIENTE-DE DAVID
```

Pero, un momento. ¿Por qué nos tomamos la molestia de introducir en nuestro programa todos los PARIENTE-DE sabiendo que alguien es pariente si es madre? Dejemos que haga el trabajo nuestro programa, mediante una regla:

```
&.METE(X PARIENTE-DE Z SI MADRE-DE Z)
        INTERPRETANDO LA REGLA
> MARIA PARIENTE-DE JUAN
        INTERPRETANDO LA REGLA
        INTERPRETANDO LA REGLA
> ANA PARIENTE-DE TERESA
        INTERPRETANDO LA REGLA
        INTERPRETANDO LA REGLA
```

```
> JUANA PARIENTE-DE LUIS
          INTERPRETANDO LA REGLA
          INTERPRETANDO LA REGLA
> ISABEL PARIENTE-DE MARTA
          INTERPRETANDO LA REGLA
```

Le proporcionamos al programa una regla similar referida a los padres, y hacemos una comprobación para ver lo que PROLOG-A ha incorporado a su base de datos al utilizar estas reglas:

```
&.LISTA TODO

MARIA MADRE-DE JUAN
JOSE PADRE-DE JUAN
ANA MADRE-DE TERESA
ALFREDO PADRE-DE TERESA
JUANA MADRE-DE LUIS
JORGE PADRE-DE LUIS
ISABEL MADRE-DE MARTA
FRANCISCO PADRE-DE MARTA
SILVIA MADRE-DE GUILLERMO
FERNANDO PADRE-DE GUILLERMO
ROSA MADRE-DE MAR
FELIX PADRE-DE MAR
TERESA MADRE-DE CRISTINA
JUAN PADRE-DE CRISTINA
MARTA MADRE-DE GERARDO
LUIS PADRE-DE GERARDO
MAR MADRE-DE ROCIO
GUILLERMO PADRE-DE ROCIO
MAR MADRE-DE ALBERTO
GUILLERMO PADRE-DE ALBERTO
ROCIO MADRE-DE ROBERTO
GERARDO PADRE-DE ROBERTO
CRISTINA MADRE-DE DAVID
ALBERTO PADRE-DE DAVID
JOSE PARIENTE-DE JUAN
ALFREDO PARIENTE-DE TERESA
JORGE PARIENTE-DE LUIS
FRANCISCO PARIENTE-DE MARTA
FERNANDO PARIENTE-DE GUILLERMO
FELIX PARIENTE-DE MAR
JUAN PARIENTE-DE CRISTINA
LUIS PARIENTE-DE GERARDO
```

```
GUILLERMO PARIENTE-DE ROCIO
GUILLERMO PARIENTE-DE ALBERTO
GERARDO PARIENTE-DE ROBERTO
ALBERTO PARIENTE-DE DAVID
MARIA PARIENTE-DE JUAN
ANA PARIENTE-DE TERESA
JUANA PARIENTE-DE LUIS
ISABEL PARIENTE-DE MARTA
SILVIA PARIENTE-DE GUILLERMO
ROSA PARIENTE-DE MAR
TERESA PARIENTE-DE CRISTINA
MARTA PARIENTE-DE GERARDO
MAR PARIENTE-DE ROCIO
MAR PARIENTE-DE ALBERTO
ROCIO PARIENTE-DE ROBERTO
CRISTINA PARIENTE-DE DAVID
```

Podemos introducirle otras relaciones, logrando así que nuestro sistema se convierta verdaderamente en un experto en lo que se refiere al árbol familiar de David:

```
&.METE(X CONOCE-A Z SI X MADRE-DE ROB
ERTO Y Z PADRE-DE DAVID)
          INTERPRETANDO LA REGLA
> ROCIO CONOCE-A ALBERTO
```

Seguidamente decimos a PROLOG-A que MARIA ESTA CASADO-CON JOSE y le revelamos el siguiente secreto:

```
&.METE(X CONYUGE-DE Y SI X CASADO-CON
 Z)
          INTERPRETANDO LA REGLA
          INTERPRETANDO LA REGLA
          INTERPRETANDO LA REGLA
> MARIA CONYUGE-DE JOSE
```

Nuestra base de datos continúa creciendo conforme aumenta el saber de PROLOG-A:

```
&.METE(X CASADO-CON Z SI X PADRE-DE L
UIS Y Z MADRE-DE LUIS)
           INTERPRETANDO LA REGLA
> JORGE CASADO-CON JUANA
```

Nuestro sistema es únicamente un sistema experto si podemos recuperar información de su base de datos. Podemos formular preguntas cortas y directas sobre un individuo:

```
.&ES(GUILLERMO PADRE-DE ROCIO)

 SI

&.CUAL(X : JOSE PADRE-DE X)

JUAN
 NO HAY (MAS) RESPUESTAS
```

O comprobar grupos relacionados:

```
&.CUAL((X Z) : X PADRE-DE Z)

JOSE JUAN
ALFREDO TERESA
JORGE LUIS
FRANCISCO MARTA
FERNANDO GUILLERMO
JUAN CRISTINA
LUIS GERARDO
GUILLERMO ROCIO
GUILLERMO ALBERTO
GERARDO ROBERTO
ALBERTO DAVID
 NO HAY (MAS) RESPUESTAS

&.CUAL((X Z) : X HIJO-DE Z)
JUAN MARIA
JUAN JOSE
 NO HAY (MAS) RESPUESTAS
```

Jugando al juego de los números

Tal como veremos a continuación, PROLOG-A dispone tanto de habilidades textuales como numéricas:

```
&.ES(SUMA(12 13 25))
 SI
```

Aquí le preguntamos al programa si la suma de los dos primeros números (12 y 13) es igual al tercero. Seguidamente vienen tres ejemplos más:

```
&.ES(SUMA(12 14 20))
 NO

&.ES(SUMA(5.4 -8 -6.3))
 NO

&.ES(SUMA(-7 -9 -16))
 SI
```

Esto tiene un uso limitado. No obstante, podemos utilizar SUMA para hallar valores desconocidos. Primeramente, sumaremos dos números:

```
&.CUAL(X : SUMA(5.4 -8 X))
-2.6
NO HAY (MAS) RESPUESTAS
```

Aquí tenemos la conocida forma "¿Cuál X es tal que resulta cierto que es igual a 12 más 32?". Alternativamente, podemos interrogar a PROLOG-A de la siguiente manera: "¿Cuál X, al sumarlo a un número conocido nos da un resultado también conocido?"

```
&.CUAL(X : SUMA(X 4 7.98))
 3.98
NO HAY (MAS) RESPUESTAS
```

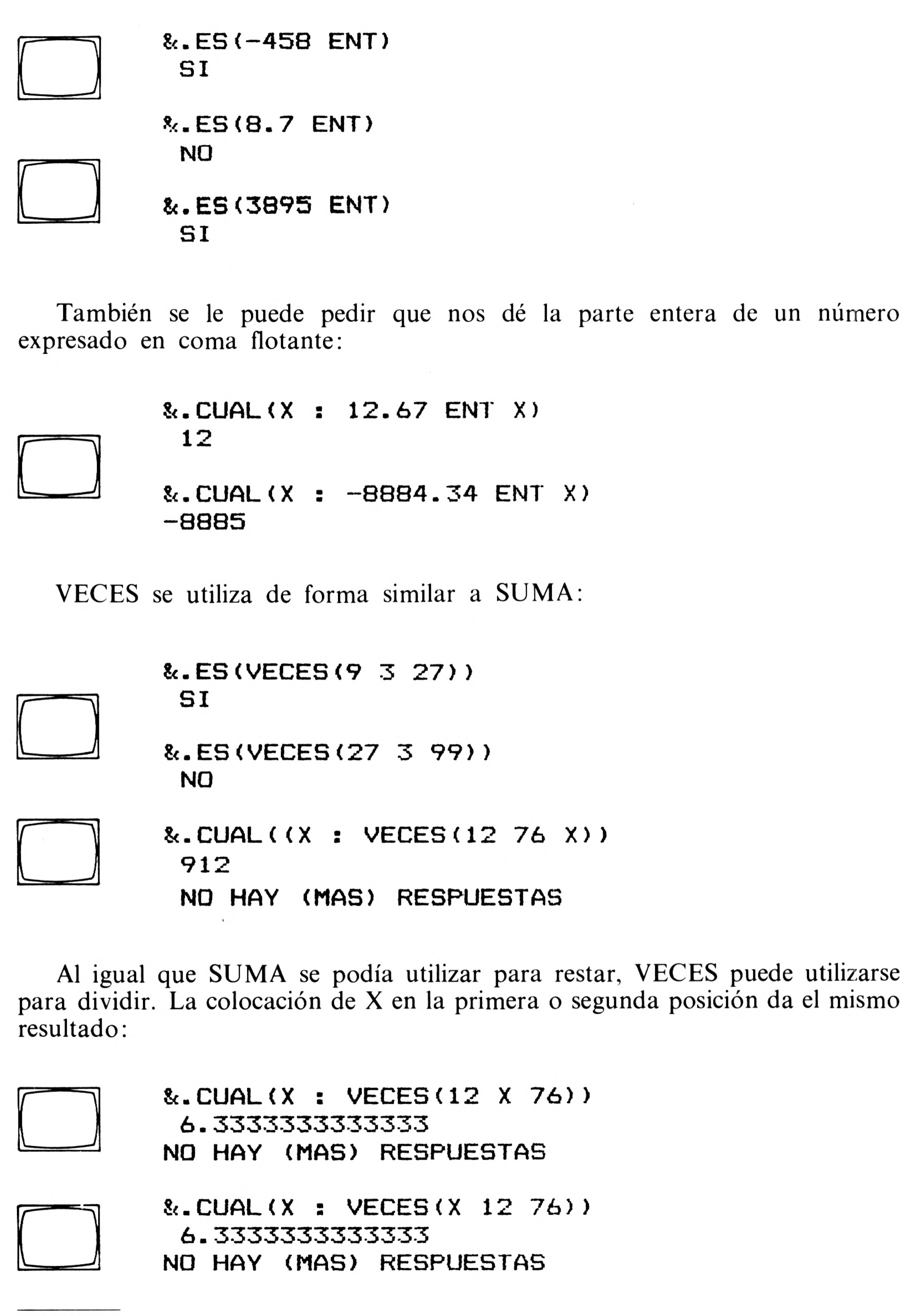
Colocando la incógnita en una posición diferente se consigue que PROLOG-A haga restas:

```
&.CUAL(X : SUMA(123 X -98))
-221
NO HAY (MAS) RESPUESTAS
```

Lo anterior equivale a preguntar "¿Cuál X es el resultado de restar el segundo número del primero?"

PROLOG-A también puede comprobar si un número es o no entero:

```
&.ES(23 ENT)
 SI
```



```
&.ES(-458 ENT)
 SI

&.ES(8.7 ENT)
 NO

&.ES(3895 ENT)
 SI
```

También se le puede pedir que nos dé la parte entera de un número expresado en coma flotante:

```
&.CUAL(X : 12.67 ENT X)
 12

&.CUAL(X : -8884.34 ENT X)
-8885
```

VECES se utiliza de forma similar a SUMA:

```
&.ES(VECES(9 3 27))
 SI

&.ES(VECES(27 3 99))
 NO

&.CUAL((X : VECES(12 76 X))
 912
 NO HAY (MAS) RESPUESTAS
```

Al igual que SUMA se podía utilizar para restar, VECES puede utilizarse para dividir. La colocación de X en la primera o segunda posición da el mismo resultado:

```
&.CUAL(X : VECES(12 X 76))
 6.3333333333333
NO HAY (MAS) RESPUESTAS

&.CUAL(X : VECES(X 12 76))
 6.3333333333333
NO HAY (MAS) RESPUESTAS
```

Aquí, PROLOG-A responde a la pregunta "¿Cuál X es tal que X es igual a 76 dividido por 12?"

El programa puede comparar la magnitud de números (y también la posición relativa de palabras, considerando como "menores que" a las que vienen primero según orden alfabético):

```
&.ES(123 MENOR 98)
 NO

&.ES(66 MENOR 144)
 NO

&.ES(YUYU MENOR DEDO)
 NO

&.ES(HOLA MENOR ADIOS)
 NO

&.ES(ESTO MENOR POR)
 NO

&.ES(A MENOR B)
 SI

&.ES(CORTO MENOR CORTISIMO)
 SI
```

De lo anterior deducirá que el trabajar con matemáticas en PROLOG-A no es nada fácil. PROLOG-A (y el LISP) no fueron diseñados en realidad para el trabajo matemático, aunque se tuvo en cuenta la realización de algunas operaciones matemáticas, y su sintaxis sigue la sintaxis del "proceso de listas", que es la verdadera función del PROLOG-A en la vida.



13 Listado de PROLOG-A

Por supuesto, PROLOG-A es, en realidad, un simulador en BASIC de PROLOG más que un verdadero intérprete, por lo cual no se comporta en todas las situaciones exactamente igual a como lo haría el frontal SIMPLE de micro-PROLOG. No obstante, vale en la mayoría de los casos (aunque la compilación de las reglas no es siempre tan perfecta como sería de desear). Con toda certeza, el programa constituye una herramienta con la que se puede aprender por sí mismo PROLOG, y mediante la cual se puede realizar gran cantidad de experimentos.

El listado que viene a continuación es bastante bueno. Para hacer PROLOG-A más asequible, lo escribí deliberadamente de modo que se pudiera omitir toda la parte *matemática*, sin afectar en absoluto al funcionamiento del programa (por supuesto que posteriormente, una vez recuperados de la introducción de la versión simplificada del PROLOG-A, podemos añadir la parte para el procesamiento numérico). Para suprimir la parte matemática basta con saltarse las líneas 2430 a 3260. También se podrían omitir (pero *no* lo recomiendo) las líneas 220, 230, 240 y 250. Si lo hacemos, habría que añadirlas posteriormente cuando se decida incorporar la parte matemática.

Tal como indica la línea 20, todas las introducciones deben efectuarse en letras mayúsculas. El programa puede manejar en su base de datos hasta 1000 líneas y compilar hasta 100 nuevos hechos en base a una regla (aunque es casi imposible que esto se necesite). Sin embargo, el programa se hace cada vez más lento (como cabía suponer) conforme la cantidad de información que mantiene

aumenta. Esto, en realidad, no importa, ya que, como veremos, el tiempo que le cuesta responder a la mayoría de las preguntas es, con todo, cuestión de segundos.

```
10 REM PROLOG-A
     ( UN SENCILLO VISTAZO )
20 REM         TODAS LAS ENTRADAS
               EN MAYUSCULAS
25 CLEAR 500
30 GOTO 50
40  PRINT "NO HAY (MAS) RESPUESTAS":R
ETURN
50 GOSUB 3270:REM INICIALIZACION
60 REM ************************
70  PRINT
80 PRINT "&.";:LINE INPUT J$
90 IF J$="" THEN 3310
95 LPRINT "&.";J$
100 IF J$="LISTA TODO" THEN GOSUB 186
0:GOTO 70
110 IF LEFT$(J$,6)="LISTA " THEN J$=M
ID$(J$,6)+" ":GOSUB 990:GOTO 70
120 IF RIGHT$(J$,1)<>")" THEN PRINT "
1.":INPUT M$:J$=J$+M$:GOTO 120
130 LJ=LEN(J$)
140 J$=LEFT$(J$,LJ-1)+" ":REM QUITA
  PARENTESIS FINAL Y LO CAMBIA POR
  UN ESPACIO
150 LJ=LEN(J$)
160 FL=0
170 IF LEFT$(J$,5)="METE(" THEN J$=MI
D$(J$,6):FL=1
180 RU=0:MA=0:AR=0
190 FOR R=1 TO LEN(J$)
200 IF MID$(J$,R,4)=" SI " THEN RU=R:
FL=6
210 IF MID$(J$,R,3)=" Y " THEN MA=R
220 IF MID$(J$,R,5)="SUMA(" THEN AR=1
230 IF MID$(J$,R,6)="VECES(" THEN AR=
2
240 IF MID$(J$,R,7)="MENOR(" THEN AR=
3
250 IF MID$(J$,R,3)="ENT" THEN AR=4
```

```
260 NEXT R
270 IF LEFT$(J$,3)="ES(" THEN J$=MID$
(J$,4):FL=2
280 IF LEFT$(J$,9)="CUAL(X : " THEN J
$=MID$(J$,10):FL=3
290 IF LEFT$(J$,15)="CUAL((X Z) : X "
 THEN J$=MID$(J$,16):FL=4
300 IF FL=0 THEN PRINT "ERROR DE SINT
AXIS":GOTO 70
310 LJ=LEN(J$)
320 REM   AHORA BUSCA LAS SUBRUTINAS
                PERTINENTES
330 IF MA<>0 THEN GOSUB 1950:GOTO 70:
REM    CODIFICA LA REGLA CONDICIONAL
               " SI "
340 IF RU<>0 AND FL<>5 THEN GOSUB 111
0:REM CODIFICA LA REGLA    " Y "
350 IF AR<>0 THEN GOSUB 2430:GOTO 70:
REM   ARITMETICA
360 IF RIGHT$(J$,3)=" X " OR RIGHT$(J
$,3)=" Z " THEN J$=LEFT$(J$,LJ-2)+" "
370 LJ=LEN(J$)
380 IF FL=1 THEN GOSUB 440:REM METE
390 IF FL=2 THEN GOSUB 520:REM ES
400 IF FL=3 THEN GOSUB 610:REM CUAL
410 IF FL=4 THEN GOSUB 830:REM CUAL2
420 GOTO 70
430 REM ************************
440 REM  METE
   ( ADQUIERE AFIRMACIONES )
450 K=0
460 K=K+1
470 IF Z$(K)="" THEN Z$(K)=J$:RETURN
480 IF K<1000 THEN 460
490 PRINT " MEMORIA COMPLETA"
500 RETURN
510 REM *************************
520 REM       ES
530 K=0
540 K=K+1
550 IF Z$(K)="" THEN 580
560 IF Z$(K)=J$ THEN LPRINT " SI":GOT
O 590
570 IF K<1000 THEN 540
580  PRINT " NO"
```

```
590 RETURN
600 REM *************************
610 REM          CUAL
      ( BUSCA LAS CORRESPONDENCIAS )
620 IF LEFT$(J$,1)="X" THEN 710
630 J$=LEFT$(J$,LJ-1)
640 K=0
650 K=K+1
660 IF Z$(K)="" THEN 690
670 IF J$=LEFT$(Z$(K),LEN(J$)) THEN P
RINT RIGHT$(Z$(K),(LEN(Z$(K))-LEN(J$)
))
680 IF K<1000 THEN 650
690 GOSUB 40
700 RETURN
710 REM * CUESTIONES QUE COMIENZAN
              CON   X  *
720 J$=MID$(J$,3,LEN(J$)-3)
730 LJ=LEN(J$)
740 K=0
750 K=K+1
760 IF Z$(K)="" THEN 800
770 Q$=MID$(Z$(K),LEN(Z$(K))-LJ,LJ)
780 IF Q$=J$ THEN LPRINT LEFT$(Z$(K),
LEN(Z$(K))-(LJ+2))
790 IF K<1000 THEN 750
800 GOSUB 40
810 RETURN
820 REM *************************
830 REM      * CUAL2 *
840 J$=LEFT$(J$,LJ-2)
850 LJ=LEN(J$)
860 K=0
870 K=K+1
880 IF Z$(K)="" THEN 960
890 LF=0
900 FOR L=1 TO LEN(Z$(K))-LJ
910 IF MID$(Z$(K),L,LJ)=J$ THEN LF=L
920 NEXT L
930 IF LF=0 THEN 950
940  PRINT LEFT$(Z$(K),LF-2);MID$(Z$(
K),LF+LJ)
950 IF K<1000 THEN 870
960 GOSUB 40
970 RETURN
```

```
980 REM ***************************
990 REM       LISTADO
1000 K=0
1010 K=K+1
1020 IF Z$(K)="" THEN RETURN
1030 LF=0
1040 FOR L=1 TO LEN(Z$(K))-LEN(J$)
1050 IF MID$(Z$(K),L,LEN(J$))=J$ THEN
 LF=1
1060 NEXT L
1070 IF LF=1 THEN  PRINT Z$(K)
1080 IF K<1000 THEN 1010
1090 RETURN
1100 REM ***********************
1110 REM  CONFORMA LAS REGLAS
1120 R=RU
1130 E$=LEFT$(J$,R):F$=MID$(J$,R+4)
1140 IF LEFT$(E$,1)<>"X" THEN  PRINT
"ERROR EN LA REGLA":GOTO 70
1150 REM LA SIGUIENTE LINEA DETECTA
 ENTRADAS COMO :
      X RELACION Z SI X ES  Z
1160 IF RIGHT$(F$,2)="Z " THEN 1390
1170  PRINT TAB(7);"INTERPRETANDO LA
REGLA"
1180 FOR T=1 TO 100
1190 R$(T)=""
1200 NEXT T
1210 E$=MID$(E$,3):F$=MID$(F$,3)
1220 K=0:RR=0
1230 K=K+1
1240 IF Z$(K)="" THEN 1300
1250 IF RIGHT$(Z$(K),LEN(F$))<>F$ THE
N 1370
1260 RR=RR+1
1270 R$(RR)=LEFT$(Z$(K),(LEN(Z$(K))-L
EN(F$)))+E$
1280  PRINT "> ";R$(RR)
1290 GOTO 1230
1300 IF RR=0 THEN RETURN
1310 RC=0
1320 RC=RC+1
1330 Z$(K)=R$(RC)
1340 IF K<1000 THEN K=K+1
1350 IF RC<RR THEN 1320
```

```
1360 RETURN
1370 IF K<1000 THEN 1230
1380 RETURN
1390 REM * REGLA CON DOS VARIABLES *
1400 FOR T=1 TO 100
1410 R$(T)=""
1420 NEXT T
1430 K=0:RR=0
1440 IF K=1000 THEN RETURN
1450 K=K+1
1460 IF Z$(K)="" THEN 1770
1470 REM  DIVISION EN TRES PALABRAS
1480 Q$=Z$(K)
1490 J=0
1500 J=J+1
1510 IF MID$(Q$,J,1)=" " THEN 1540
1520 IF J<LEN(Q$) THEN 1500
1530  PRINT "ERROR EN LA INTERPRETACI
ON DE REGLA ":GOTO 70
1540 A$=LEFT$(Q$,J)
1550 Q$=MID$(Q$,J+1)
1560 J=0
1570 J=J+1
1580 IF MID$(Q$,J,1)=" " THEN 1610
1590 IF J<LEN(Q$) THEN 1570
1600  PRINT "ERROR EN LA INTERPRETACI
ON DE REGLA ":GOTO 70
1610 B$=LEFT$(Q$,J)
1620 Q$=MID$(Q$,J+1)
1630 J=0
1640 J=J+1
1650 IF MID$(Q$,J,1)=" " THEN 1680
1660 IF J<LEN(Q$) THEN 1640
1670 PRINT "ERROR EN LA INTERPRETACIO
N DE REGLA ":GOTO 70
1680 PRINT TAB(17);"INTERPRETANDO LA
REGLA"
1690 C$=LEFT$(Q$,J)
1700 M$=MID$(F$,3,LEN(B$))
1710 IF B$<>M$ THEN 1440
1720 RR=RR+1
1730 N$=MID$(E$,3,LEN(E$)-4)
1740 R$(RR)=A$+N$+C$
1750 PRINT "> ";R$(RR)
1760 GOTO 1440
```

```
1770 IF RR=0 THEN RETURN
1780 M=0
1790 M=M+1
1800 IF M>RR THEN RETURN
1810 Z$(K)=R$(M)
1820 IF K=1000 THEN PRINT " NO ESTA E
N MEMORIA":GOTO 70
1830 K=K+1
1840 GOTO 1790
1850 REM ************************
1860 REM      LISTA TODO
1870 PRINT
1880 K=0
1890 K=K+1
1900 IF Z$(K)="" THEN RETURN
1910 PRINT Z$(K)
1920 IF K<1000 THEN 1890
1930 RETURN
1940 REM ***********************
1950 REM CONFORMA LAS REGLAS CON "Y"
  DEL SIGUIENTE TIPO :
1960 REM ( X ABSORBE Z SI X ES UNA
 COSA Y Z ES OTRA COSA )
1970 REM EL CONCEPTO X DEBE ESTAR EN
  LISTA, PRECEDIENDO A Z PARA TODOS
  LOS EJEMPLOS QUE SE CODIFIQUEN
1980 REM DIVIDIDOS EN SECCIONES
1990 J$=MID$(J$,2):REM ELIMINA "X"
2000 PRINT TAB(10);"INTERPRETANDO LA
REGLA"
2010 J=1
2020 J=J+1
2030 IF MID$(J$,J,1)=" " THEN 2060
2040 IF J<LEN(J$) THEN 2020
2050 PRINT "ERROR EN LA INTERPRETACIO
N DE REGLA":RETURN
2060 A$=LEFT$(J$,J):REM RELACION 1
2070 J$=MID$(J$,J+7):REM DIVIDE AL
 COMIENZO DE LA SEGUNDA RELACION
2080 J=1:CN=0
2090 J=J+1
2100 IF MID$(J$,J,1)=" " THEN CN=CN+1
2110 IF CN=2 THEN 2140
2120 IF J<LEN(J$) THEN 2090
2130 PRINT "ERROR EN LA INTERPRETACIO
```

```
N DE REGLA":RETURN
2140 C$=MID$(J$,J):REM CONCEPTO 1
2150 C$=MID$(J$,J+4):REM CONCEPTO 2
2160 IF C$=" " THEN PRINT "ERROR EN L
A INTERPRETACION DE REGLA":RETURN
2170 REM  AHORA VA A LA BASE DE DATOS
2180 FOR T=1 TO 200
2190 R$(T)=""
2200 NEXT T
2210 R1=0:R2=99
2220 K=0
2230 K=K+1
2240 IF Z$(K)="" THEN 2310
2250 IF R1=99 OR R2=200 THEN PRINT "
FALTA MEMORIA":GOTO 2310
2260 LB=LEN(B$)
2270 IF RIGHT$(Z$(K),LB)=B$ THEN R1=R
1+1:R$(R1)=LEFT$(Z$(K),LEN(Z$(K))-LB)
2280 LC=LEN(C$)
2290 IF RIGHT$(Z$(K),LC)=C$ THEN R2=R
2+1:R$(R2)=LEFT$(Z$(K),LEN(Z$(K))-LC)
2300 IF K<1000 THEN 2230
2310 IF R$(100)="" THEN PRINT "EL SEG
UNDO CONCEPTO NO ESTA EN LA    BASE D
E DATOS ":RETURN
2320 REM AHORA CODIFICA LAS REGLAS
2330 R1=0:R2=99
2340 R2=R2+1
2350 R1=R1+1
2360 IF R$(R1)>" " AND R$(R2)>" " THE
N Z$(K)=R$(R1)+A$+R$(R2)+" "
2370 PRINT "> ";Z$(K)
2380 K=K+1
2390 IF R$(R2+1)<>"" THEN 2340
2400 IF R$(R1+1)<>"" THEN 2350
2410 RETURN
2420 REM ************************
2430 REM ARITMETICA
2440 LJ=LEN(J$)
2450 IF AR<3 THEN GOSUB 2490
2460 IF AR=3 THEN GOSUB 2890
2470 IF AR=4 THEN GOSUB 3080
2480 RETURN
2490 REM ***** SUMA - VECES *****
2500 J$=MID$(J$,5,LJ-5)
```

```
2510 IF LEFT$(J$,2)="S(" THEN J$=MID$
(J$,3) ELSE J$=MID$(J$,2)
2520 LJ=LEN(J$)
2530 K=0
2540 K=K+1
2550 IF MID$(J$,K,1)=" " THEN A$=LEFT
$(J$,K-1):J$=MID$(J$,K+1):GOTO 2580
2560 IF K<LJ THEN 2540
2570 PRINT TAB(12);"ERROR ARITMETICO"
:RETURN
2580 LJ=LEN(J$)
2590 K=0
2600 K=K+1
2610 IF MID$(J$,K,1)=" " THEN B$=LEFT
$(J$,K-1):J$=MID$(J$,K+1):GOTO 2640
2620 IF K<LJ THEN 2600
2630 PRINT TAB(12);"ERROR ARITMETICO"
:RETURN
2640 LJ=LEN(J$)
2650 K=0
2660 K=K+1
2670 IF MID$(J$,K,1)=")" THEN C$=LEFT
$(J$,K-1):GOTO 2700
2680 IF K<LJ THEN 2660
2690 PRINT TAB(4);"ERROR-(DEMASIADAS
VARIABLES)":RETURN
2700 AN=0:BN=0:CN=0
2710 IF ASC(A$)>58 THEN AN=1
2720 IF ASC(B$)>58 THEN BN=2
2730 IF ASC(C$)>58 THEN CN=4
2740 GU=AN+BN+CN:IF GU=3 OR GU=5 OR G
U=6 THEN 2690
2750 IF AR=2 THEN 2820:REM VECES
2760 IF GU>0 THEN 2790
2770 IF VAL(A$)+VAL(B$)=VAL(C$) THEN
PRINT " SI":RETURN
2780 PRINT " NO":RETURN
2790 IF GU=1 THEN PRINT VAL(C$)-VAL(B
$):GOSUB 40:RETURN
2800 IF GU=2 THEN PRINT VAL(C$)-VAL(A
$):GOSUB 40:RETURN
2810 PRINT VAL(A$)+VAL(B$):GOSUB 40:R
ETURN
2820 REM *  VECES  *
2830 IF GU>0 THEN 2860
```

```
2840 IF VAL(A$)*VAL(B$)=VAL(C$) THEN
PRINT " SI":RETURN
2850 PRINT " NO":RETURN
2860 IF GU=1 THEN PRINT VAL(C$)/VAL(B
$):GOSUB 40:RETURN
2870 IF GU=2 THEN PRINT VAL(C$)/VAL(A
$):GOSUB 40:RETURN
2880 PRINT VAL(A$)*VAL(B$):GOSUB 40:R
ETURN
2890 REM ****** MENOR ***********
2900 NF=0
2910 IF ASC(J$)<58 THEN NF=1:REM
          SON NUMEROS
2920 CN=0
2930 K=0
2940 K=K+1
2950 IF MID$(J$,K,1)=" " THEN CN=CN+1
2960 IF CN=2 THEN 3000
2970 IF K<LEN(J$) THEN 2940
2980 PRINT TAB(10);"ERROR EN LA COMPA
RACION"
2990 RETURN
3000 B$=MID$(J$,K+1)
3010 A$=LEFT$(J$,K-6)
3020 IF NF=1 THEN 3050
3030 IF A$<B$ THEN PRINT " SI":RETURN
3040 PRINT " NO":RETURN
3050 REM ***** NUMEROS *****
3060 IF VAL(A$)<VAL(B$) THEN PRINT "
SI":RETURN
3070 PRINT " NO":RETURN
3080 REM ***** ENT **********
3090 IF RIGHT$(J$,2)="X " THEN 3190
3100 K=0
3110 K=K+1
3120 IF MID$(J$,K,1)=" " THEN 3160
3130 IF K<LEN(J$) THEN 3110
3140 PRINT TAB(10);"ERROR ARITMETICO"
3150 RETURN
3160 A=VAL(LEFT$(J$,K-1))
3170 IF INT(A)=A THEN PRINT " SI":RET
URN
3180 PRINT " NO":RETURN
3190 K=0
3200 K=K+1
```

```
3210 IF MID$(J$,K,1)=" " THEN 3240
3220 IF K<LEN(J$) THEN 3200
3230 PRINT TAB(10);"ERROR ARITMETICO"
:RETURN
3240 PRINT INT(VAL(LEFT$(J$,K-1)))
3250 RETURN
3260 REM ***********************
3270 REM      INICIALIZACION
3280 CLS:KEYOFF
3290 DIM Z$(1000),R$(200)
3300 RETURN
3310 CLS
3320 LOCATE 3,10
3330 PRINT "QUIERES TERMINAR ? ( S ,
N )"
3340 A$=INKEY$
3350 IF A$="S" THEN CLS:KEYON:END
3360 IF A$="N" THEN CLS:GOTO 70
3370 GOTO 3340
```



14
Un poderoso LISP

El PROLOG que hemos estudiado en los capítulos anteriores es un descendiente del LISP (al igual que los lenguajes SMALLTALK y LOGO). El aprendizaje de la utilización del LISP es mucho más difícil que el de la del frontal SIMPLE del PROLOG, pero esta dificultad viene acompañada de un significativo aumento de flexibilidad y de potencia. Una vez que hayamos dominado el PROLOG-A, estaremos en disposición de introducirnos en el LISP.

Las palabras

Solamente hay dos tipos de palabras en LISP, átomos y listas. Un átomo es una palabra (o una combinación de letras y números, que comienza por una letra). No puede contener espacios ni cualquier carácter que no sea un número o una letra. Las palabras siguientes son átomos:

ORDENADOR CHORIZO R2D2 C3P0

Una lista se compone de átomos y otras listas. Comienza con un paréntesis a la izquierda, seguido de átomos y listas, y acaba con un paréntesis a la derecha. También puede haber otros pares de paréntesis en el interior de los dos exteriores. La lista más simple contiene un único átomo, como: (ORDENADOR), o más de un átomo, separados por espacios: (ORDENADOR CHORIZO R2D2). LISP admite una lista vacía, que es precisamente ( ). A ésta se le llama *lista nula.*

Una lista puede contener a otras listas. Consideremos la lista (COMO LO HACE). Podría estar *dentro* de otra lista: (LA PREGUNTA ES (COMO LO HACE)). Las listas (y la proliferación de paréntesis) pueden hacerse complejas, como en el caso de la lista ((COMO (LO (HACE)) AHORA) DON JOSE). Así, el gran número de paréntesis ha dado lugar a la teoría de que LISP, en vez de ser las siglas de *LISt Processing* (Procesador de listas), significa *Lots of Infuriatingly Stupid Parentheses* (cantidad de exasperantes paréntesis estúpidos).

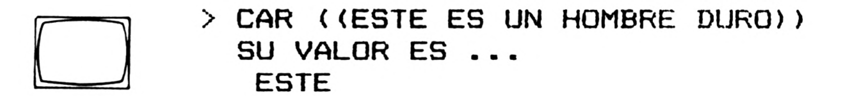
No se preocupe si esto parece ya hacerse complejo. Nuestros programas tipo LISP, EASLE y SSLISP le llevarán a través del manejo de listas de forma relativamente cómoda, haciendo cosas tales como el contar por usted el número de paréntesis, y asegurarse de que están equilibrados.

Las funciones básicas

En LISP tiene seis funciones básicas. Nuestro primer programa, EASLE (*Easy And Simple Lisp-like Exerciser:* Ejercitador tipo Lisp fácil y sencillo), nos permitirá experimentar con ellas. Las seis funciones son CAR, CDR, CONS, ATOM, NULO e IGUAL. En un sistema LISP real, se pueden utilizar precisamente estas seis para crear casi cualquier función de ordenador que pueda ser creada. Pronto aprenderemos lo que hacen estas seis funciones, viéndolas en acción en EASLE. (Ambos, EASLE y SSLISP, siguen la sintaxis de EUQ-LISP, que es una de las más fáciles de utilizar. Otros dialectos, como EU-LISP o P-LISP, exigen una sintaxis ligeramente distinta. Sin embargo, todos los LISP, especialmente en su nivel básico, son bastante parecidos, y las experiencias que adquiera al trabajar con mis programas pueden adaptarse fácilmente a las exigencias de cualquier sistema que acabe utilizando.)

CAR

Se utiliza para conseguir el primer elemento de una lista. La función se utiliza con un par de paréntesis propios que encierran la lista que se quiere comprobar (que a su vez está encerrada en su propio par de paréntesis). El caso más simple es como el siguiente:



En el ejemplo siguiente, el primer elemento de la lista es otra lista, (ESTE ES), como puede ver cuando use CAR:

```
> CAR (((ESTE ES) UN HOMBRE DURO))
  SU VALOR ES ...
   (ESTE ES)
```

Donde PROLOG respondía SI y NO a preguntas, LISP devuelve un C-CIERTO (por "verdad") o un NADA (por "falso"). Una lista vacía, ( ), es considerada por el sistema como NADA, tal como veremos en el ejemplo siguiente, en el que la lista vacía es el primer elemento de la lista a la que aplicamos la función CAR:

```
> CAR ((() (ESTE ES UN DURO)))
  SU VALOR ES ...
  NADA
```

Anteriormente dijimos que EASLE y SSLISP cuentan los paréntesis por cuenta del usuario. Lo podemos ver en el ejemplo siguiente, en el que accidentalmente se ha omitido el paréntesis final ")":

```
> CAR ((ESTE ES) (EL MEJOR)
 -> OMISION DE   )
    +    )
  SU VALOR ES ...
   (ESTE ES)
```

CDR

Esta función devuelve la lista que queda después de aplicar la función CAR a una lista. Lo veremos en los ejemplos siguientes:

```
> CDR ((NACIO EN PISCIS))
  SU VALOR ES ...
   (EN PISCIS)
```

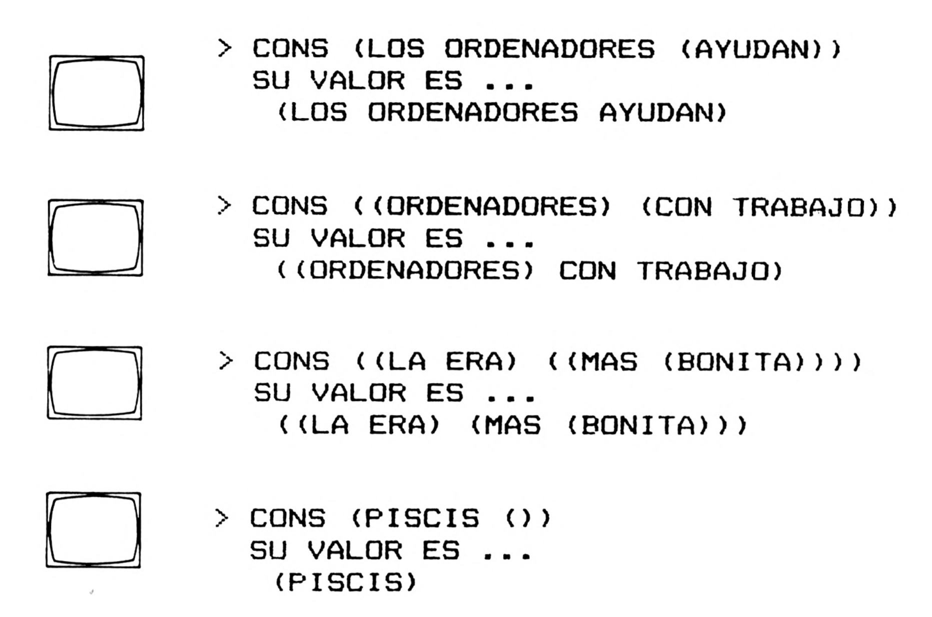
```
> CDR (((EL NACIO EN) PISCIS))
  SU VALOR ES ...
   (PISCIS)
```

```
> CDR (((LA MEJOR EPOCA)))
  SU VALOR ES ...
  NADA
```

```
> CDR ((() (LA ERA ESPECTRAL)))
  SU VALOR ES ...
   ((LA ERA ESPECTRAL))
```

CONS

Esta función une listas (al contrario que CAR y CDR, que las separan). Combina dos *argumentos*, el segundo de los cuales debe ser una lista, tal y como se muestra en la siguiente prueba del funcionamiento del programa EASLE:



Podemos ver que una lista a la que ha aplicado la función CONS se compone de un CAR y un CDR.

ATOM

Después de ver las funciones CAR, CDR y CONS, vamos a tratar ahora la función ATOM, que es mucho más sencilla que las tres anteriores. Anteriormente dije que, cuando se hace una interrogación, LISP contesta con un C-CIERTO (por "verdad") o con NADA (por "falso" o "lista vacía"). ATOM (junto con las siguientes funciones que veremos, IGUAL y NULO) son funciones de *comprobación*, y devuelven un C-CIERTO o un NADA. ATOM devuelve un C-CIERTO si el único elemento considerado de la lista es un átomo. Los ejemplos siguientes aclararán lo anterior:

```
> ATOM (AVION)
  SU VALOR ES ...
   C - CIERTO
```

```
> ATOM ((LA ERA DE PISCIS))
  SU VALOR ES ...
    NADA
```

```
> ATOM ((PISCIS))
  SU VALOR ES ...
    NADA
```

IGUAL

Vimos anteriormente que ATOM devuelve un C-CIERTO si el único argumento es un átomo, y NADA en cualquier caso. IGUAL devuelve un C-CIERTO si los dos argumentos de IGUAL son átomos idénticos; en otro caso proporciona un NADA.

Veamos a IGUAL en acción:

```
> IGUAL (PISCIS PECES)
  SU VALOR ES ...
    NADA
```

```
> IGUAL (PECERA PECERA)
  SU VALOR ES ...
   C - CIERTO
```

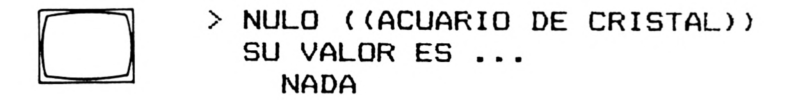
NULO

Finalmente, llegamos a la sexta función, una función de comprobación llamada NULO. Esta posee un solo argumento y devuelve un C-CIERTO si el argumento es la lista vacía ( ) y NADA en caso contrario:

```
> NULO (())
  SU VALOR ES ...
   C - CIERTO
```

```
> NULO ((()))
  SU VALOR ES ...
    NADA
```

```
> NULO ()
  SU VALOR ES ...
    * NULO NECESITA ARGUMENTO *
```



EASLE

El verdadero LISP (no así EASLE) permite introducir expresiones compuestas complejas para evaluarlas, tales como:

CONS (CAR((FRANCIA)) (CONS(NOS AMA)
CDR ((RGA CON TERNURA))))

que da como resultado:

SU VALOR ES...
(FRANCIA NOS AMA CON TERNURA)

SSLISP (que viene de *Single Statement LISP:* Sentencias Sencillas LISP) no admite expresiones múltiples, lo que limita fuertemente su valor como herramienta por propio derecho. No obstante, aunque EASLE y SSLISP no pueden utilizarse tal y como son para crear sistemas expertos, sirven con toda seguridad para proporcionar al usuario las ideas necesarias para enfrentarse con un LISP completo.

Ahora, cargue y ejecute EASLE, y utilícelo para evaluar varias listas por sí mismo:

```
10 REM        EASLE
20 CLS:KEYOFF
30 DIM X(40),Y(40),Z(40)
40 PRINT
50 PRINT "> ";:LINE INPUT A$
60 IF A$="" THEN 970 :REM PULSA
    < RETURN > PARA TERMINAR
70 REM **************************
80 FOR J=1 TO 40
90 X(J)=0:Y(J)=0:Z(J)=0
100 NEXT J
110 REM **************************
120 R=0:S=0:T=0:CUNO=0:CDOS=0:ED=0
130 FOR J=1 TO LEN(A$)
140 B$=MID$(A$,J,1)
150 IF B$="(" THEN S=S+1:Z(S)=J:IF T=
0 THEN CUNO=J
```

```
160 IF B$=")" THEN T=T+1:Y(T)=J:IF CD
OS<>0 AND ED=0 THEN ED=J-1
170 IF T=1 AND B$=")" THEN CDOS=J
180 IF B$=" " THEN R=R+1:X(R)=J
190 NEXT J
200 IF S=T THEN 260:REM EQUILIBRADO
        DE PARENTESIS  ( )
210 IF S<T THEN PRINT " -> OMISION DE
  ("
220 IF S>T THEN PRINT " -> OMISION DE
  )"
230 PRINT "    +    ";:LINE INPUT B$
240 A$=A$+B$
250 GOTO 80
260 FLAG=0:B$=" NADA"
270 IF LEFT$(A$,5)="CAR (" THEN FLAG=
1
280 IF LEFT$(A$,5)="CDR (" THEN FLAG=
2
290 IF LEFT$(A$,6)="CONS (" THEN FLAG
=3
300 IF LEFT$(A$,6)="ATOM (" THEN FLAG
=4
310 IF LEFT$(A$,7)="IGUAL (" THEN FLA
G=5
320 IF LEFT$(A$,6)="NULO (" THEN FLAG
=6
330 ON FLAG GOSUB 420,470,550,690,780
,920
340 IF FLAG<>0 THEN GOSUB 360
350 GOTO 40
360 REM ** CONTESTACION DE RETORNO **
370 PRINT "  SU VALOR ES ..."
380 IF B$<>"()" THEN PRINT "   ";B$
390 IF B$="()" THEN PRINT "  NADA"
400 RETURN
410 REM *************************
420 REM     **   CAR   **
430 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2
)-Z(2))
440 IF S>2 THEN B$=MID$(A$,CUNO,CDOS-
CUNO+1)
450 RETURN
460 REM *************************
470 REM     **   CDR   **
```

```
480 GOSUB 420
490 LB=LEN(B$)+7
500 B$="("+MID$(A$,LB,ED-1)
510 IF RIGHT$(B$,2)="))" THEN B$=LEFT
$(B$,LEN(B$)-1)
520 IF MID$(B$,2,1)=" " THEN B$="("+M
ID$(B$,3)
530 RETURN
540 REM ************************
550 REM      **   CONS   **
560 B$=MID$(A$,7,LEN(A$)-1)
570 J=0
580 IF LEFT$(B$,1)="(" THEN J=1
590 J=J+1
600 IF MID$(B$,J,1)="(" THEN 630
610 IF J<LEN(B$) THEN 590
620 B$="  * CONS  INCORRECTO *":RETUR
N
630 LB=LEN(B$)-1
640 B$="("+LEFT$(B$,J-1)+MID$(B$,J+1)
650 B$=LEFT$(B$,LB)
660 IF RIGHT$(B$,2)=" )" THEN B$=LEFT
$(B$,LEN(B$)-2)+")"
670 RETURN
680 REM **********************
690 REM      **   ATOM   **
700 A$=MID$(A$,7,LEN(A$)-1)
710 J=0
720 J=J+1
730 IF MID$(A$,J,1)=" " OR MID$(A$,J,
1)="(" THEN RETURN
740 IF J<LEN(A$) THEN 720
750 B$="C - CIERTO"
760 RETURN
770 REM **********************
780 REM        **  IGUAL   **
790 A$=MID$(A$,8)
800 A$=LEFT$(A$,LEN(A$)-1)
810 J=0
820 J=J+1
830 IF MID$(A$,J,1)=")" THEN RETURN
840 IF MID$(A$,J,1)=" " THEN 870
850 IF J<LEN(A$) THEN 820
860 RETURN
870 C$=LEFT$(A$,J-1)
```

```
880 A$=MID$(A$,J+1)
890 IF C$=A$ THEN B$="C - CIERTO"
900 RETURN
910 REM **********************
920 REM   **   NULO   **
930 IF A$="NULO ()" THEN PRINT " * NU
LO NECESITA ARGUMENTO *"
940 B$=" "
950 IF A$="NULO (())" THEN B$="C - CI
ERTO"
960 RETURN
970 CLS:KEYON
```



15 SSLISP

Con EASLE hemos aprendido a utilizar las seis funciones básicas de LISP: CAR, CDR, CONS, ATOM, IGUAL y NULO. Con el siguiente programa, que es más parecido al LISP, el SSLISP (de Sentencias Sencillas LISP), tendremos acceso a estas seis funciones originales y otras más, hasta un total de treinta y una, que proporcionan la mayoría de los sistemas LISP.

SSLISP nos permitirá incluso definir nuestras propias funciones, así que, si, por ejemplo, pensamos que HUMO es un nombre más lógico que CAR, podemos definir una función de nombre HUMO, que se comporte exactamente igual a como lo hace CAR. SSLISP nos autoriza a definir hasta veinte funciones propias.

Dado que las funciones son exactamente las mismas en SSLISP que en EASLE, no daremos ninguna muestra adicional de su funcionamiento. En vez de ello, nos concentraremos en las nuevas funciones.

MIEMBRO

Esta función busca la presencia de la primera lista en el resto de la lista completa, devolviendo la lista de la cual la primera lista es el resultado de la función CAR. Si no encuentra tal lista, devuelve un NADA. MIEMBRO utiliza la función MISMO (que veremos a su debido tiempo) en su definición, mientras que la siguiente función —MIGUAL— utiliza IGUAL. (En algunos sistemas

MIEMBRO comprueba la presencia de un átomo en el interior de una lista, dando como resultado C-CIERTO o NADA:

```
> MIEMBRO ((EL FINAL) (TRAS (EL FINAL
) (NOS FUIMOS)))
  SU VALOR ES ...
   ((EL FINAL) (NOS FUIMOS))
```

```
> MIEMBRO (ESTO ((ESTO ES) UN TEST))
  SU VALOR ES ...
    NADA
```

MIGUAL

Como dijimos anteriormente, MIGUAL utiliza en su definición IGUAL, donde MIEMBRO utilizaba MISMO. El siguiente ejemplo muestra el efecto, en la práctica, de este cambio:

```
> MIGUAL (AHORA (QUIZAS AHORA ESTEMOS
 MEJOR))
  SU VALOR ES ...
   (AHORA ESTEMOS MEJOR)
```

```
> MIGUAL (AHORA (QUIZAS (AHORA) SEAN
MAS))
  SU VALOR ES ...
    NADA
```

JUNTA

Su mismo nombre sugiere que JUNTA se utiliza para crear una única lista uniendo otras dos. El ejemplo siguiente nos la muestra en acción:

```
> JUNTA ((VAMOS AL) (COLEGIO))
  SU VALOR ES ...
  (VAMOS AL COLEGIO)
```

```
> JUNTA ((HACE DOS) (EN UNA))
  SU VALOR ES ...
  (HACE DOS EN UNA)
```

INVIERTE

Esta función es ideal para poner al revés las tablas. Invierte los elementos de una lista, excepto los encerrados en sus propios paréntesis. Lo aclaramos con dos ejemplos:

```
> INVIERTE ((ARRIBA ABAJO ATRAS))
  SU VALOR ES ...
  (ATRAS ABAJO ARRIBA)
```

```
> INVIERTE ((NADIE SABE (COMO (GANARL
E))))
  SU VALOR ES ...
  ( (COMO (GANARLE)) NADIE SABE)
```

MISMO

Hace poco que mencionamos esta función al discutir MIEMBRO y MIGUAL. MISMO comprueba si las dos partes de la lista que está analizando son idénticas. Como podemos ver, los dos argumentos de esta función pueden ser números, listas o átomos:

```
> MISMO ((ESTE ES) (ESTE ES))
  SU VALOR ES ...
   C - CIERTO
```

```
> MISMO ((ESTE SI) (ESTE NO))
  SU VALOR ES ...
    NADA
```

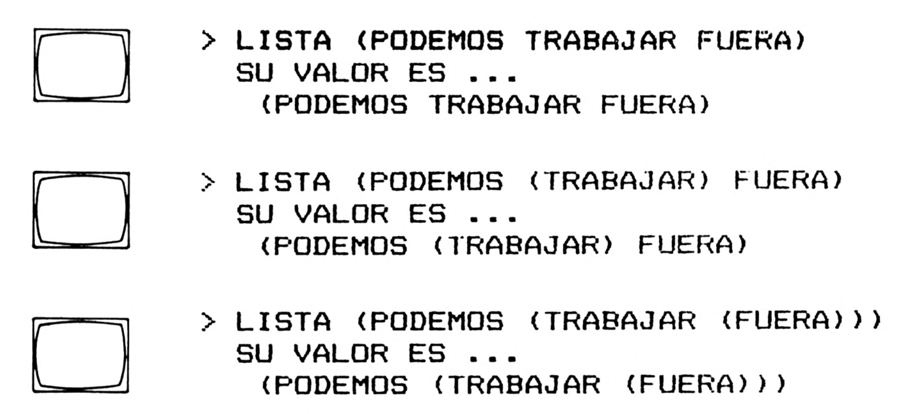
```
> MISMO ((ESE (ES)) (ESE (ES)))
  SU VALOR ES ...
   C - CIERTO
```

```
> MISMO (9 9)
  SU VALOR ES ...
   C - CORRECTO
```

```
> MISMO (9 -9)
  SU VALOR ES ...
    NADA
```

LISTA

Mientras que MISMO, y muchas otras funciones de LISP, pueden tomar dos y sólo dos argumentos, LISTA aceptará cualquier número de ellos. Da como salida los valores de los argumentos de la función (es decir, simplemente los lista):



```
> LISTA (PODEMOS TRABAJAR FUERA)
  SU VALOR ES ...
   (PODEMOS TRABAJAR FUERA)
```

```
> LISTA (PODEMOS (TRABAJAR) FUERA)
  SU VALOR ES ...
   (PODEMOS (TRABAJAR) FUERA)
```

```
> LISTA (PODEMOS (TRABAJAR (FUERA)))
  SU VALOR ES ...
   (PODEMOS (TRABAJAR (FUERA)))
```

Operaciones matemáticas

El primer par de funciones que estudiaremos son INC1 y DEC1, las cuales tienen la función de sumar o restar, respectivamente, una unidad al argumento que las sigue:

```
> INC1 (9)
  SU VALOR ES ...
    10

> INC1 (-100)
  SU VALOR ES ...
   -99

> INC1 (0)
  SU VALOR ES ...
    1

> DEC1 (0)
  SU VALOR ES ...
-1

> DEC1 (-99)
  SU VALOR ES ...
   -100
```

La función CERO? comprueba si el argumento es un cero o no, devolviendo o NADA o C-CIERTO. La función UNO? hace la misma comprobación para el número uno:

```
> CERO? (1)
  SU VALOR ES ...
    NADA

> CERO? (9)
  SU VALOR ES ...
    NADA

> CERO? (0)
  SU VALOR ES ...
   C - CORRECTO

> UNO? (1)
  SU VALOR ES ...
   C - CORRECTO

> UNO? (0)
  SU VALOR ES ...
    NADA
```

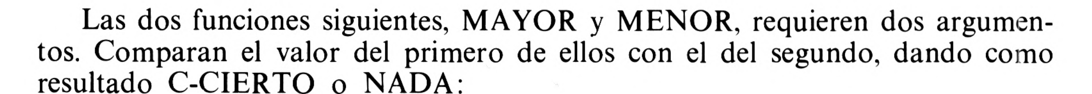
NUMERO? responderá C-CIERTO si el argumento es un número, o NADA si no lo es, mientras que NEGATIVO? comprobará si el argumento es un número negativo o no:

```
> NUMERO? (OCHO)
  SU VALOR ES ...
    NADA

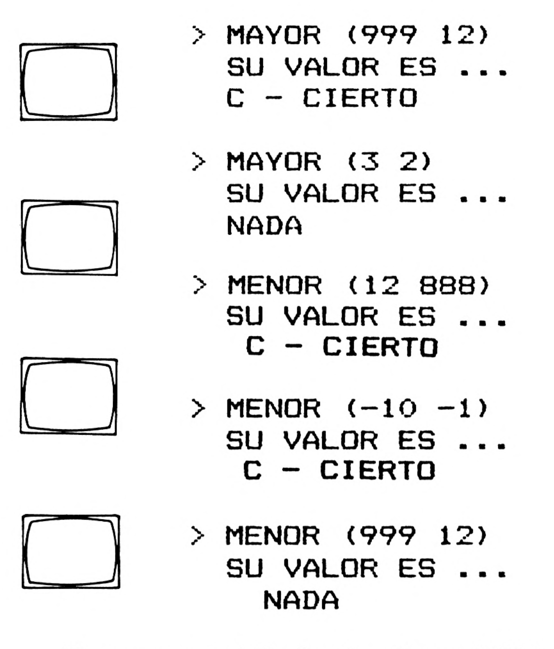
> NUMERO? (9)
  SU VALOR ES ...
   C - CIERTO)

> NEGATIVO? (7)
  SU VALOR ES ...
    NADA

> NEGATIVO? (-7)
  SU VALOR ES ...
   C - CIERTO
```



Las dos funciones siguientes, MAYOR y MENOR, requieren dos argumentos. Comparan el valor del primero de ellos con el del segundo, dando como resultado C-CIERTO o NADA:



```
> MAYOR (999 12)
  SU VALOR ES ...
  C - CIERTO

> MAYOR (3 2)
  SU VALOR ES ...
  NADA

> MENOR (12 888)
  SU VALOR ES ...
   C - CIERTO

> MENOR (-10 -1)
  SU VALOR ES ...
   C - CIERTO

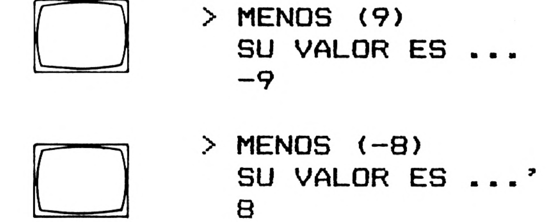
> MENOR (999 12)
  SU VALOR ES ...
    NADA
```

Tenemos también las funciones MX (máximo) y MIN (mínimo), que pueden aceptar cualquier número de argumentos y dan como resultado, respectivamente, el valor máximo y el mínimo encontrados en sus respectivas listas:

```
> MAX (9 -8 123 0 33 19)
  SU VALOR ES ...
    123

> MIN (9 -8 123 0 33 19)
  SU VALOR ES ...
    167
```

MENOS cambia el signo de su único argumento (multiplicándolo por menos uno):



```
> MENOS (9)
  SU VALOR ES ...
  -9

> MENOS (-8)
  SU VALOR ES ...
  8
```

Operando numéricamente

Mientras que las anteriores funciones matemáticas son esencialmente "pasivas", produciendo una declaración relativa a sus argumentos o modificándolos en una unidad, el grupo de funciones siguiente se utiliza para realizar propiamente operaciones matemáticas.

RESTA calcula la diferencia en valor entre sus dos argumentos, restando el segundo del primero:

```
> RESTA (34 12)
  SU VALOR ES ...
    22
```

```
> RESTA (100 -100)
  SU VALOR ES ...
    200
```

Con la función EXP (exponencial) podemos elevar el primer argumento a la potencia indicada por el segundo:

```
> EXP (3 3)
  SU VALOR ES ...
    27
```

```
> EXP (3 -3)
  SU VALOR ES ...
    .037037037037037
```

RECIP calcula el valor de uno partido por el argumento:

```
> RECIP (3)
  SU VALOR ES ...
 .33333333333333
```

```
> RECIP (10)
  SU VALOR ES ...
 .1
```

```
> RECIP (.1)
  SU VALOR ES ...
 10
```

DIVIDE divide el primer argumento por el segundo:

```
> DIVIDE (9 8)
  SU VALOR ES ...
   1.125
```

```
> DIVIDE (3 12)
  SU VALOR ES ...
 .25
```

RESTO devuelve el resto que queda después de haber calculado el cociente de dos números (en muchas versiones de BASIC, el operador MOD devuelve el valor entero que es el resto de una división entera: RESTO es su equivalente en LISP:

```
> RESTO (10 3)


  SU VALOR ES ...
    1
```

```
> RESTO (100 9)
  SU VALOR ES ...
    1
```

```
> RESTO (99 11)
  SU VALOR ES ...
    0
```

```
> RESTO (12)
  SU VALOR ES ...
    ERROR - NO SE ADMITE UN UNICO
            ARGUMENTO
```

```
> RESTO (12 3)
  SU VALOR ES ...
    0
```

Finalmente, mientras que muchas de las funciones anteriores solamente pueden manejar dos argumentos, las dos siguientes, SUMA y MULT, aceptan cualquier número de ellos:

```
> SUMA (3 -8 5)
  SU VALOR ES ...
 0

> SUMA (13 -8 4)
  SU VALOR ES ...
 9

> MULT (24 .1)
  SU VALOR ES ...
    2.4

> MULT (24 .01)
  SU VALOR ES ...
    .24

> MULT (12 3 3)
  SU VALOR ES ...
    108
```



16
Definición de funciones LISP por el usuario

Al discutir las seis funciones básicas disponibles en los sistemas LISP (CAR, CDR y las demás) mencionamos que las podíamos utilizar para definir cualquier función que pudiera ser creada en un sistema ordenador. LISP utiliza la función DEFINE para definir nuevas funciones. La función DEFINE que vamos a analizar sigue la sintaxis estándar de LISP para la definición de funciones por el usuario, por lo que la experiencia que adquirimos con SSLISP podremos aplicarla directamente a un sistema LISP completo. No obstante, debido a que SSLISP sólo puede aceptar una sentencia única cada vez, lo único que podemos hacer aquí es cambiar el nombre de funciones que el sistema soporta. Esto nos permitirá cambiar los nombres de hasta veinte de las funciones que nos proporciona el SSLISP.

La sintaxis para la utilización de DEFINE es la siguiente:

```
DEFINE (NUEVO-NOMBRE(LAMBDA(VARIABLE-FICTICIA)(FUNCION
                    VARIABLE-FICTICIA)))
```

La definición de esta función, lo mismo que la de las restantes funciones de LISP, es una lista. La función CAR de esta lista es el nombre de la función (NUEVO-NOMBRE en el ejemplo anterior). Este nombre puede ser cualquier átomo, pero no debe coincidir con el nombre utilizado ya por el sistema para una de sus funciones. El nombre viene seguido de la palabra LAMBDA (que viene del formalismo lógico denominado cálculo-Lambda desarrollado y ex-

puesto por Alonso Church en su *Introduction to Mathematical Logic*, vol. 1, Princeton, New Jersey, Princeton University Press, 1956). A pesar de su impresionante linaje, podemos ignorarla, aunque recordando que debe ponerse (al igual que en el resto de los sistemas LISP del mundo) para que la función DEFINE trabaje.

Después de LAMBDA viene una variable ficticia del tipo sobre el cual esperamos que nuestra palabra función definida opere. Seguidamente viene la verdadera función del sistema que deseamos sea imitada por nuestro NUEVO-NOMBRE. (En sistemas LISP reales, se pueden utilizar aquí otras definiciones, lo cual lleva a poder crear complejas y poderosas definiciones. FORTH otorga el mismo poder.) El último elemento de la lista es de nuevo una variable ficticia.

Todo esto puede parecer un poco lioso. (Yo también la encontré como uno de los elementos más difíciles de comprender de LISP.) Sin embargo, una vez lo hayamos visto en acción, nos daremos cuenta de que es bastante fácil.

Como ya sabemos, la función CAR devuelve el *primer* elemento de una lista. Nuestra primera utilización de DEFINE consistirá en crear una función denominada HUMO que nos devolverá el resultado de aplicar la función CAR a una lista. Para ello introducimos lo siguiente en nuestro sistema SSLISP:

```
> DEFINE (HUMO (LAMBDA (DER) (CAR DER
)))
  SU VALOR ES ...
   HUMO
```

Ahora la comprobamos utilizando HUMO como si fuera una función estándar del sistema:

```
> HUMO ((ESTO ES UN TEST))
  SU VALOR ES ...
   ESTO
```

```
> HUMO (((ESTE ES) UN HOMBRE DURO))
  SU VALOR ES ...
   (ESTE ES)
```

¡Funciona! Si queremos utilizar para multiplicar una X en vez de la palabra MULT, podemos definir X de forma que:

```
> DEFINE (X (DER (NUM) (MULT NUM)))
  SU VALOR ES ...
   X
```

```
> X (8 9)
  SU VALOR ES ...
    72
```

Podemos hacer que el BOMBO se comporte igual que MAX:

```
> DEFINE (BOMBO (DFR (ER) (MAX ER)))
  SU VALOR ES ...
   BOMBO

> BOMBO (1 44 -9 182 12)
  SU VALOR ES ...
    182
```

Para acabar añadiremos SOLO a nuestro vocabulario, y haremos que equivalga a ATOM:

```
> DEFINE (SOLO (SAS (WERT) (ATOM WERT
)))
  SU VALOR ES ...
   SOLO

> SOLO (PALABRA)
  SU VALOR ES ...
   C - CIERTO
```



17 Listado del SSLISP

A continuación viene el listado completo de SSLISP para que pueda probarlo en su ordenador:

```
10 REM   S.S.LISP
20 GOSUB 3440:REM INICIALIZACION
30 REM        TODAS LAS ENTRADAS EN
                 MAYUSCULAS
40 A$=MID$(A$,E):A$=LEFT$(A$,LEN(A$)-
1)
50 RETURN
60 REM ****************************
70 PRINT
80 NN=0
90 PRINT "> ";:LINE INPUT A$
100 IF A$="" THEN 3500:REM PULSA
     < RETURN > PARA TERMINAR
110 REM ****************************
120 FOR J=1 TO 12
130 X(J)=0:Y(J)=0:Z(J)=0
140 NEXT J
```

```
150 REM ***************************
160 R=0:S=0:T=0:CUNO=0:CDOS=0:ED=0
170 FOR J=1 TO LEN(A$)
180 B$=MID$(A$,J,1)
190 IF B$="(" THEN S=S+1:Z(S)=J:IF T=
0 THEN CUNO=J
200 IF B$=")" THEN T=T+1:Y(T)=J:IF CD
OS<>0 AND ED=0 THEN ED=J-1
210 IF T=1 AND B$=")" THEN CDOS=J
220 IF B$=" " THEN R=R+1:X(R)=J
230 NEXT J
240 IF S=T THEN 300:REM EQUILIBRADO
        DE PARENTESIS  ( )
250 IF S<T THEN PRINT " -> OMISION DE
  ("
260 IF S>T THEN PRINT " -> OMISION DE
  )"
270 PRINT "     +    ";:LINE INPUT B$
280 A$=A$+B$
290 GOTO 120
300 IF NWDS=0 OR NN=1 THEN 370
310 M$=LEFT$(A$,X(1)-1)
320 FOR J=1 TO NWDS
330 IF M$=N$(J) THEN A$=O$(J)+MID$(A$
,LEN(N$(J))+1)
340 NEXT J
350 NN=1
360 GOTO 120
370 FLAG=0:B$=" NADA"
380 IF LEFT$(A$,5)="CAR (" THEN FLAG=
1
390 IF LEFT$(A$,5)="CDR (" THEN FLAG=
2
400 IF LEFT$(A$,6)="CONS (" THEN FLAG
=3
410 IF LEFT$(A$,6)="ATOM (" THEN FLAG
=4
420 IF LEFT$(A$,7)="IGUAL (" THEN FLA
G=5
430 IF LEFT$(A$,6)="NULO (" THEN FLAG
=6
440 IF LEFT$(A$,9)="MIEMBRO (" THEN F
LAG=7
450 IF LEFT$(A$,8)="MIGUAL (" THEN FL
AG=8
```

```
460 IF LEFT$(A$,7)="JUNTA (" THEN FLA
G=9
470 IF LEFT$(A$,10)="INVIERTE (" THEN
 FLAG=10
480 IF LEFT$(A$,7)="MISMO (" THEN FLA
G=11
490 IF LEFT$(A$,7)="LISTA (" THEN FLA
G=12
500 IF LEFT$(A$,8)="DEFINE (" THEN FL
AG=13
510 IF LEFT$(A$,6)="INC1 (" THEN FLAG
=14
520 IF LEFT$(A$,6)="DEC1 (" THEN FLAG
=15
530 IF LEFT$(A$,7)="CERO? (" THEN FLA
G=16
540 IF LEFT$(A$,7)="RESTA (" THEN FLA
G=17
550 IF LEFT$(A$,5)="EXP (" THEN FLAG=
18
560 IF LEFT$(A$,5)="MAX (" THEN FLAG=
19
570 IF LEFT$(A$,5)="MIN (" THEN FLAG=
20
580 IF LEFT$(A$,6)="SUMA (" THEN FLAG
=21
590 IF LEFT$(A$,7)="MENOS (" THEN FLA
G=22
600 IF LEFT$(A$,8)="DIVIDE (" THEN FL
AG=23
610 IF LEFT$(A$,7)="RECIP (" THEN FLA
G=24
620 IF LEFT$(A$,7)="RESTO (" THEN FLA
G=25
630 IF LEFT$(A$,6)="MULT (" THEN FLAG
=26
640 IF LEFT$(A$,7)="MAYOR (" THEN FLA
G=27
650 IF LEFT$(A$,7)="MENOR (" THEN FLA
G=28
660 IF LEFT$(A$,11)="NEGATIVO? (" THE
N FLAG=29
670 IF LEFT$(A$,9)="NUMERO? (" THEN F
LAG=30
680 IF LEFT$(A$,6)="UNO? (" THEN FLAG
```

```
=31
690 IF FLAG>13 THEN 720
700 ON FLAG GOSUB 840,890,970,1110,12
00,1330,1380,1510,1700,1760,2000,2130
,3300
710 GOTO 760
720 IF FLAG>24 THEN 750
730 ON FLAG-13 GOSUB 2180,2230,2280,2
350,2560,2790,2820,2870,2920,2960
740 GOTO 760
750 ON FLAG-24 GOSUB 3030,3070,3120,3
160,3200,3250,2280
760 IF FLAG<>0 THEN GOSUB 780
770 GOTO 70
780 REM ** CONTESTACION DE RETORNO **
790 PRINT "  SU VALOR ES ..."
800 IF B$<>"()" THEN PRINT "    ";B$
810 IF B$="()" THEN PRINT "  NADA"
820 RETURN
830 REM *************************
840 REM     **   CAR   **
850 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2
)-Z(2))
860 IF S>2 THEN B$=MID$(A$,CUNO,CDOS-
CUNO+1)
870 RETURN
880 REM *************************
890 REM     **   CDR   **
900 GOSUB 840
910 LB=LEN(B$)+7
920 B$="("+MID$(A$,LB,ED-1)
930 IF RIGHT$(B$,2)="))" THEN B$=LEFT
$(B$,LEN(B$)-1)
940 IF MID$(B$,2,1)=" " THEN B$="("+M
ID$(B$,3)
950 RETURN
960 REM *************************
970 REM     **   CONS   **
980 B$=MID$(A$,7,LEN(A$)-1)
990 J=0
1000 IF LEFT$(B$,1)="(" THEN J=1
1010 J=J+1
1020 IF MID$(B$,J,1)="(" THEN 1050
1030 IF J<LEN(B$) THEN 1010
1040 B$="  * CONS  INCORRECTO *":RETU
```

```
RN
1050 LB=LEN(B$)-1
1060 B$="("+LEFT$(B$,J-1)+MID$(B$,J+1
)
1070 B$=LEFT$(B$,LB)
1080 IF RIGHT$(B$,2)=" )" THEN B$=LEF
T$(B$,LEN(B$)-2)+")"
1090 RETURN
1100 REM **********************
1110 REM      **   ATOM   **
1120 A$=MID$(A$,7,LEN(A$)-1)
1130 J=0
1140 J=J+1
1150 IF MID$(A$,J,1)=" " OR MID$(A$,J
,1)="(" THEN RETURN
1160 IF J<LEN(A$) THEN 1140
1170 B$="C - CIERTO"
1180 RETURN
1190 REM **********************
1200 REM        **  IGUAL   **
1210 E=8:GOSUB 40
1220 J=0
1230 J=J+1
1240 IF MID$(A$,J,1)=")" THEN RETURN
1250 IF MID$(A$,J,1)=" " THEN 1280
1260 IF J<LEN(A$) THEN 1230
1270 RETURN
1280 C$=LEFT$(A$,J-1)
1290 A$=MID$(A$,J+1)
1300 IF C$=A$ THEN B$="C - CIERTO"
1310 RETURN
1320 REM **********************
1330 REM    **   NULO   **
1340 IF A$="NULO ()" THEN B$=" * NULO
 NECESITA ARGUMENTO *"
1350 IF A$="NULO (())" THEN B$="C - C
IERTO"
1360 RETURN
1370 REM **********************
1380 REM     **   MIEMBRO    **
1390 C$=MID$(A$,10)
1400 J=1
1410 J=J+1
1420 IF MID$(C$,J,1)=")" OR MID$(C$,J
,1)="(" THEN D$=LEFT$(C$,J):GOTO 1450
```

```
1430 IF J<LEN(C$) THEN 1410
1440 RETURN
1450 J=LEN(D$)
1460 J=J+1
1470 IF MID$(C$,J,LEN(D$))=D$ THEN C$
=LEFT$(C$,LEN(C$)-1):GOTO 1630
1480 IF J<LEN(C$) THEN 1460
1490 RETURN
1500 REM ************************
1510 REM     **   MIGUAL   **
1520 C$=MID$(A$,9)
1530 J=0
1540 J=J+1
1550 IF MID$(C$,J,1)=" " THEN 1580
1560 IF J<LEN(C$) THEN 1540
1570 RETURN
1580 D$=LEFT$(C$,J)
1590 C$=MID$(C$,J+2)
1600 C$=LEFT$(C$,LEN(C$)2)+" "
1610 J=0
1620 J=J+1
1630 IF MID$(C$,J,LEN(D$))=D$ THEN B$
="("+MID$(C$,J):GOTO 1660
1640 IF J<LEN(C$) THEN 1620
1650 RETURN
1660 B$=LEFT$(B$,LEN(B$)-1)+")"
1670 IF RIGHT$(B$,3)=")))" THEN B$=LE
FT$(B$,LEN(B$)-1):GOTO 1670
1680 RETURN
1690 REM ***********************
1700 REN     **   JUNTA   **
1710 B$=MID$(B$,8)
1720 B$=LEFT$(B$,Y(1)-8)+" "+MID$(B$,
Z(3)-6)
1730 B$=LEFT$(B$,LEN(B$)-1)
1740 RETURN
1750 REM **********************
1760 REM      ** INVIERTE  **
1770 B$=""
1780 A$=MID$(A$,11):A$=LEFT$(A$,LEN(A
$)-2)
1790 CT=0
1800 J=0
1810 J=J+1:IF J>LEN(A$) THEN 1920
1820 IF MID$(A$,J,1)=" " THEN 1850
```

```
1830 IF MID$(A$,J,1)="(" THEN 1860
1840 GOTO 1810
1850 CT=CT+1:G$(CT)=LEFT$(A$,J-1):A$=
MID$(A$,J+1):GOTO 1800
1860 J=J+1:IF MID$(A$,J,2)="))" THEN
1980
1970 IF MID$(A$,J,1)=")" THEN 1910
1880 IF J=LEN(A$) THEN 1900
1890 GOTO 1860
1900 CT=CT+1:G$(CT)=A$+")":GOTO 1930
1910 CT=CT+1:G$(CT)=LEFT$(A$,J):A$=MI
D$(A$,J+1):GOTO 1800
1920 CT=CT+1:G$(CT)=A$
1930 FOR M=CT TO 1 STEP -1
1940 B$=B$+G$(M):IF M>1 THEN B$=B$+"
"
1950 NEXT M
1960 B$="("+B$+")"
1970 RETURN
1980 CT=CT+1:G$(CT)=LEFT$(A$,J+1):A$=
MID$POKEA$,J+2):GOTO 1800
1990 REM ***********************
2000 REM    **   MISMO   **
2010 E=8:GOSUB 40
2020 M=ASC(A$):IF M>47 AND M<58 THEN
2370
2030 J=0
2040 J=J+1
2050 IF MID$(A$,J,2)=") " THEN J=J+1:
GOTO 1280
2060 IF MID$(A$,J,3)=")))" THEN 2100
2070 IF MID$(A$,J,2)="))" THEN 2110
2080 IF J<LEN(A$) THEN 2040
2090 RETURN
2100 C$=LEFT$(A$,J+2):A$=MID$(A$,J+4)
:GOTO 1300
2110 C$=LEFT$(A$,J+1):A$=MID$(A$,J+3)
:GOTO 1300
2120 REM ************************
2130 REM      **   LISTA   **
2140 E=8:GOSUB 40
2150 B$="("+A$+")"
2160 RETURN
2170 REM ************************
2180 REM      **   INC1   **
```

```
2190 E=7:GOSUB 40
2200 B$=STR$(VAL(A$)+1)
2210 RETURN
2220 REM *************************
2230 REM      **    DEC1    **
2240 E=7:GOSUB 40
2250 B$=SRT$(VAL(A$)-1)
2260 RETURN
2270 REM *************************
2280 REM      **    CERO?    **
2290 IF FLAG=16 THEN E=8
2300 IF FLAG=31 THEN E=7
2310 GOSUB 40
2320 IF A$="0" AND FLAG=16 OR A$="1"
AND FLAG=31 THEN B$="C - CIERTO"
2330 RETURN
2340 REM *************************
2350 REM    ** DOS ARGUMENTOS **
2360 E=8:GOSUB 40
2370 J=0
2380 J=J+1
2390 IF MID$(A$,J,1)=" " THEN 2420
2400 IF J<LEN(A$) THEN 2380
2410 B$=" ERROR - NO SE ADMITE UN UNI
CO                    ARGUMENTO ":RETURN
2420 P=VAL(LEFT$(A$,J-1))
2430 Q=VAL(MID$(A$,J+1))
2440 IF FLAG=17 THEN B$=STR$(P-Q):RET
URN
2450 IF FLAG=23 OR FLAG=25 THEN B=P/Q
2460 IF FLAG=25 THEN B=INT(.5+Q*(B-IN
T(B))*1000)/1000
2470 IF FLAG=18 THEN B=P^Q
2480 IF FLAG=11 AND P=Q THEN B$="C -
CIERTO"
2490 IF FLAG=27 AND P>Q THEN B$="C -
CIERTO"
2500 IF FLAG=28 AND P<Q THEN B$="C -
CIERTO"
2510 IF FLAG=32 THEN B=P-Q
2520 IF FLAG=11 OR FLAG>26 THEN RETUR
N
2530 B$=STR$(B)
2540 RETURN
2550 REM *************************
```

```
2560 REM     ***   EXP   ***
2570 E=6:GOSUB 40
2580 GOTO 2370
2590 REM *************************
2600 REM  ** MAX MIN SUMA MULT **
2610 F$=LEFT$(A$,3):A$=MID$(A$,6)
2620 CT=0:FLAG=0
2630 IF F$="MULT" THEN CT=1
2640 J=0
2650 J=J+1
2660 IF MID$(A$,J,1)=" " THEN 2690
2670 IF J<LEN(A$) THEN 2650
2680 IF J=LEN(A$) THEN FLAG=1
2690 P=VAL(LEFT$(A$,J-1)):IF FLAG=0 T
HEN A$=MID$(A$,J+1)
2700 IF F$<>"SUMA" AND CT=0 THEN CT=P
2710 IF F$="MAX" AND P>CT THEN CT=P
2720 IF F$="MIN" AND P<CT THEN CT=P
2730 IF F$="SUMA" THEN CT=CT+P
2740 IF F$="MULT" THEN CT=CT*P
2750 IF FLAG=0 THEN 2640
2760 B$=STR$(CT)
2770 RETURN
2780 REM ************************
2790 REM      **   MIN   **
2800 GOTO 2610
2810 REM ************************
2820 REM      **   SUMA   **
2830 F$="SUMA"
2840 A$=MID$(A$,7)
2850 GOTO 2620
2860 REM ************************
2870 REM         **  MENOS   **
2880 E=8:GOSUB 40
2890 B$=STR$(-VAL(A$))
2900 RETURN
2910 REM ************************
2920 REM      **  DIVIDE   **
2930 E=9:GOSUB 40
2940 GOTO 2370
2950 REM ************************
2960 REM      **  RECIP  **
2970 E=8:GOSUB 40
2980 IF A$="0" THEN B$=" DIVISION POR
 CERO ; ILEGAL ":RETURN
```

```
2990 B=1/(VAL(A$))
3000 B$=STR$(B)
3010 RETURN
3020 REM ************************
3030 REM       **  RESTO  **
3040 E=8:GOSUB 40
3050 GOTO 2370
3060 REM ************************
3070 REM     **   MULT   **
3080 F$="MULT"
3090 A$=MID$(A$,7)
3100 GOTO 2620
3110 REM ************************
3120 REM      **   MAYOR   **
3130 E=8:GOSUB 40
3140 GOSUB 2370
3150 REM ************************
3160 REM      **   MENOR   **
3170 E=8:GOSUB 40
3180 GOTO 2370
3190 REM ************************
3200 REM   **   NEGATIVO?   **
3210 E=12:GOSUB 40
3220 IF VAL(A$)<0 THEN B$="C - CIERTO
 "
3230 RETURN
3240 REM ************************
3250 REM    **   NUMERO?   **
3260 A$=MID$(A$,10)
3270 IF ASC(A$)>44 AND ASC(A$)<58 THE
N B$="C - CIERTO) "
3280 RETURN
3290 REM ************************
3300 REM      ** DEFINE  **
3310 A$=MID$(A$,9)
3320 F$=LEFT$(A$,X(2)-9)
3330 G$=MID$(A$,X(4)-6)
3340 J=0
3350 J=J+1
3360 IF MID$(G$,J,1)=" " THEN 3390
3370 IF J<LEN(G$) THEN 3350
3380 B$=" ERROR EN LA DEFINICION ":RE
TURN
3390 G$=LEFT$(G$,J-1)
3400 NWDS=NWDS+1
```

```
3410 O$(NWDS)=G$:N$(NWDS)=F$
3420 B$=F$
3430 RETURN
3440 REM ************************
3450 REM  INICIALIZACION
3460 CLS:KEYOFF
3470 DIM G$(20),O$(20),N$(20),X(12),Y
(12),Z(12)
3480 NWDS=0:REM CONTADOR DE NUEVAS
 PALABRAS DEFINIDAS POR EL USUARIO
3490 GOTO 70
3500 CLS
3510 LOCATE 3,10
3520 PRINT "QUIERES TERMINAR ?  ( S,N
 )"
3530 A$=INKEY$
3540 IF A$="S" THEN CLS:END
3550 IF A$="N" THEN CLS:GOTO 70
3560 GOTO 3530
```

Apéndice A

Algunos comentarios sobre los lenguajes declarativos LISP y PROLOG

Definición de lenguaje de programación

El glosario desarrollado por la Federación Internacional para el Tratamiento de la Información (FITI), y publicado con el título *Vocabulary of Information Processing* por North-Holland Publishing, Co. (Amsterdam, Holanda, 1966), en su página 79 define a la palabra *lenguaje* como "término general asociado a un conjunto de *símbolos* definidos y un conjunto de reglas o convenciones que gobiernan la forma en que se pueden combinar dichos símbolos para conseguir un proceso de comunicación significativo". Acompaña a esta definición la nota siguiente: "Un lenguaje que no es ambiguo, pensado para expresar *programas*, se denomina *lenguaje de programación.*"

Los lenguajes de programación y sus dialectos se cuentan por centenares, si no por millares. Los lenguajes naturales de comunicación humana quizá les superen en número, pero en determinados aspectos los lenguajes de programación divergen todavía más. Cada lenguaje tiene su gramática y su sintaxis, así como la manera de expresar las ideas, y, en principio, dependiendo de la tarea a realizar o el problema a resolver, la labor de escribir un programa se facilita enormemente seleccionando determinados lenguajes en vez de otros.

Clasificación de los lenguajes de programación

No es fácil clasificar los lenguajes de programación, ya que un lenguaje determinado puede pertenecer a las dos categorías que se van a indicar a continuación, pero sirven para clarificar las ideas en la torre de Babel que forman los actuales lenguajes de programación.

En informática se utiliza la palabra *variable* para hacer referencia a una localización de memoria a la que se le ha asignado un nombre. El contenido de la localización de memoria es el valor de la variable.

Los lenguajes de programación pueden actuar de dos formas muy diferentes sobre las variables, y en función de esta actuación vamos a clasificarlos en lenguajes imperativos y lenguajes declarativos.

Lenguajes imperativos

Estos lenguajes ofrecen un conjunto específico de operaciones ejecutables y de sentencias, entre las que destaca *la de asignación*, que permiten modificar los valores de las variables. Mediante una adecuada organización de dichas sentencias se obtiene el programa que resuelve un problema determinado. Ejemplos característicos de lenguajes imperativos son: BASIC, FORTRAN, COBOL, PASCAL, C, etc.

Lenguajes declarativos

Estos lenguajes tienen la propiedad conocida como transparencia referencial, es decir, prohíbe asignar valores diferentes a una variable durante la ejecución del programa. Por tanto, las variables son tales en el sentido de que de una ejecución a otra pueden variar, pero una vez comenzada la ejecución son variables estables.

Cuando se utilizan lenguajes imperativos se especifica el procedimiento para resolver cada problema, mientras que con los lenguajes declarativos se especifica qué tipo de solución se busca. Por ejemplo, para calcular el cubo del número 10 con un lenguaje imperativo, se inicializa la variable CUBO a 1 y, después, mediante un bucle formado por 3 lazos, se hace CUBO igual a CUBO por 10. Con un lenguaje declarativo se debería definir qué significa cubo y después preguntar: ¿Cuál es el cubo de 10?

Dos ejemplos característicos de lenguajes declarativos son el LISP y el PROLOG.

LISP

Este es el más antiguo de los lenguajes declarativos. Desde la época en que se diseñó por John McCarthy en el Instituto Tecnológico de Massachusetts han aparecido numerosas versiones con características imperativas, pero a pesar de ellos sigue manteniendo la posibilidad de actuar como lenguaje declarativo puro.

La estructuración estándar de datos es la lista y el lenguaje ofrece funciones de selección como *car* y *cdr*, funciones de construcción como *list* y *cons* y predicados como *null*.

Mediante lo que se denomina expresión lambda se puede definir y manipular funciones.

La unidad básica en un programa LISP es la expresión, y cada una calcula un valor.

La recursión es el único mecanismo de control.

Su historia puede resumirse de la forma siguiente: el trabajo se inició en 1959 por el Grupo de Inteligencia Artificial de MJT, bajo la dirección del profesor John McCarthy; la versión inicial se publicó en marzo de 1960 y se implementó en un JBM 704. Posteriormente se desarrolló la versión 1.5 para el JBM 709. La razón para este esquema de numeración se supone que fue debido a que los cambios y mejoras efectuadas no eran como para denominarla versión 2. Como resultado de esta política fueron apareciendo sucesivas versiones: 1.75, 1.9, etc.

El lenguaje LISP no es muy usual debido a que no está bien preparado para cualquier cosa que no sea manipulación simbólica, procesos recursivos y procesado de listas. Es extremadamente difícil de leer y de escribir debido a la existencia de una gran cantidad de paréntesis, que lo convierte en excesivamente propenso a los errores. No está pensado para personas sin o con poca experiencia, ya que requiere una base ligeramente sofisticada para apreciar y utilizar su eficiencia.

Actualmente las versiones existentes son independientes de la máquina.

La aportación más significativa de todas las versiones anteriores a la 2 es su contribución al desarrollo de la teoría general de la programación.

LISP 2 se creó con objeto de resolver los problemas que tenían las versiones anteriores, y que son las siguientes:

— Aritmética muy lenta, haciendo impracticable la utilización del LISP para cálculo.
— Notación muy desventajosa por exceso de paréntesis.
— Necesidad de escribir todas las expresiones matemáticas en notación polaca.

Por todo ello, en 1963 un grupo del Instituto Tecnológico de Massachusetts se puso a trabajar en una versión más eficiente. La versión LISP 2 se diferencia fundamentalmente en que el lenguaje fuente se basa en el ALGOL 60, con algunos de los conceptos del LISP añadidos.

Según sus diseñadores, algunas áreas de aplicación típicas para el LISP 2 son: manipulación algebraica, análisis lingüístico, inteligencia artificial, reconocimiento de estructuras, cálculo numérico, etc. Excluyendo el campo de la gestión, parece ser, por lo dicho, que este lenguaje es el más ambicioso de los creados hasta la fecha, aunque parece poco probable que en alguna de las áreas indicadas pueda llegar a desbancar a otros lenguajes fuertemente introducidos.

A pesar de estar basado en ALGOL, el lenguaje no es una extensión del ALGOL, aunque sus conocedores encontrarán el aprendizaje del LISP 2 facilitado en gran medida. En la actualidad no está todavía extendido.

Bibliografía

En el conjunto de referencias dadas se han agrupado, por su orientación:

— Manuales de definición realizados por los diseñadores:

1. McCarthy, J., y otros: *LISP 1.5 Programmer's Manual*, M. J. T. Computation Center and Research Laboratory of Electronics, Cambridge, Mass. (agosto 1962).

— Publicaciones de introducción al proceso de listas con discusión de los conceptos de listas:

1. Wilkes, M. V.: «Lists and why they are useful", Proc. ACM 19th Nat. L Conf., 1964, pp. 1-15.
2. Woodeard, P. M.: "List programming", en *Advances in Programming and Non-Numerical Computation* (L. Fox, ed.), Pergamon Press, Nueva York, 1966, pp. 29-48.

— Libros sencillos introductorios:

1. Berkeley, E. C., y Bobrow, D. G.: *The Programming Language LISP. Its Operation and Applications*, M. J. T. Press, Cambridge, Mass., 1966.
2. Weissman, C.: *LISP 1.5 Primer*, Dickenson Publishing Co., Belmont, California, 1967.

LISP 2

— Manuales de definición:

1. Abrahams, P. W., y otros: *The LISP 2 Programming Language and System*, Proc. FJCC, vol. 29 (1966), pp. 661-676.

— Libros de introducción:

1. Levin, M., y Berkeley, E.: *LISP 2 Primer*, System Development Corp., TM-2710/101/00 (draft), Santa Mónica, California (julio 1966).

PROLOG

Este lenguaje fue inventado en Francia y mejorado en Inglaterra, aunque existe actualmente la polémica sobre su paternidad, que también se atribuye un grupo de investigadores húngaros. En la actualidad se le puede considerar como un lenguaje incompleto, pues muchas versiones carecen de controladores de periféricos, se quedan cortas en cuanto a capacidad de cálculo, pues tan sólo utiliza los enteros positivos y las operaciones elementales (suma, resta, multiplicación y división), y carecen de control de errores.

Desde 1972 han aparecido varias implementaciones (Marsella, Edimburgo, Imperial College, Waterloo, etc.), cada una con diferente sintaxis.

A diferencia de otros lenguajes en los que los programas se forman mediante conjuntos de funciones, un programa PROLOG está formado por una sucesión de relaciones y reglas relativas a un sujeto. Todo ello forma una base de datos de información que se puede consultar o aumentar.

PROLOG es el acrónimo de PROgramming in LOGic. La programación lógica debe su origen al desarrollo de la lógica en general y a los avances de la lógica matemática en particular.

En la década de los cincuenta, los lógicos inclinados hacia la informática comenzaron a investigar técnicas para automatizar las demostraciones de los teoremas matemáticos.

Dos hitos tuvieron lugar a mitad de los años sesenta. Alan Robinson desarrolló la regla de resolución por inferencia y Donald Loveland desarrolló el modelo para la prueba por eliminación. Hasta 1970, ambos métodos, que estaban expresados mediante notaciones diferentes, permanecieron completamente desconectados. Ese año, Donald Kuchner demostró que el modelo de eliminación y una forma de resolución denominada "lineal" podrían ser contempladas como equivalentes. Robert Kowalski y sus colaboradores desarrollaron una síntesis de ambos métodos que denominaron resolución SL. De forma independiente, pero al mismo tiempo, Donald Loveland y Raymon Reiter desarrollaron métodos de prueba similares.

En 1972, Alain Colmeraner y Phillippe Roussel, en Marsella, diseñaron e implementaron PROLOG como un desarrollo de la resolución SL, construyendo un intérprete escrito en ALGOL. A partir de ese momento se han diseñado versiones mejoradas para distintos equipos.

La popularidad del PROLOG se ha incrementado enormemente desde que los japoneses anunciaron que su utilización sería uno de los principales elementos en su proyecto de construcción de los ordenadores de la quinta generación, que incorporarán los lenguajes tipo PROLOG al nivel de los actuales "lenguajes máquina".

Bibliografía

Clark, K., y McCabe, F. G.: *Micro-PROLOG: Programming in Logic*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

Ejemplos

Con objeto de ilustrar la forma de actuación de un lenguaje de programación del tipo declarativo y mostrar la propiedad de transparencia referencial, vamos a usar un ejemplo muy conocido e indicaremos la actuación del PROLOG en este caso.

Supongamos que tenemos las siguientes afirmaciones dadas en el orden siguiente:

Turing es humano	A1
Sócrates es humano	A2
Sócrates es griego	A3
X es falible si X es humano	A4

Y supongamos que tenemos que resolver el problema de encontrar un griego falible planteado de la forma siguiente:

Y es falible e Y es griego?

PROLOG trabajaría de forma automática resolviendo sucesivamente los siguientes subproblemas:

— Teniendo en cuenta A4 transforma la pregunta en

Y es humano e Y es griego?

— Teniendo en cuenta A1 y, además, como no hay condiciones en A1, transforma la pregunta en

Turing es griego?

Como no puede resolver esto, vuelve a la pregunta anterior:

Y es humano e Y es griego?

Utiliza A2 y de forma análoga se plantea:

Sócrates es griego?

PROLOG resuelve el problema definitivamente con la afirmación A3 y, por tanto, realiza la deducción Y = Sócrates.

Apéndice B

Teorema de Bayes y probabilidades

El reverendo Thomas Bayes, que vivió de 1702 a 1761, fue un ministro presbiteriano y un cualificado matemático. Los sistemas expertos apenas podían soñarse en su tiempo (y sus técnicas se hubieran atribuido probablemente al diablo). A pesar de ello, sus trabajos encuentran rápidas aplicaciones en cualquier campo (como el de los sistemas expertos) donde las probabilidades de que ocurran ciertos sucesos tienen que modificarse conforme se acumula información adicional.

Es de sospechar que Bayes no podía estar satisfecho. Inició el camino que conduce al desarrollo de su teorema al expresar la admirable hipótesis de que la existencia del Todopoderoso podía demostrarse mediante el examen de las bellezas matemáticas existentes en el mundo que El había creado. "Probaré —dijo— que el Principio Final de la Divina Providencia... es la Felicidad de Sus Criaturas, y lo haré mediante las matemáticas."

Desafortunadamente, cuanto más avanzaba en sus estudios, más se alarmaba por las implicaciones de sus descubrimientos. Finalmente, cerró el libro sobre su trabajo, y decidió que no podía publicarlo en su época.

Sin embargo, el trabajo que hizo era sólido y descansa en el corazón de la moderna teoría de las decisiones. A menudo se le llama, en su honor, teoría de la decisión de Bayes. Su teorema nos proporciona un medio matemático adecuado de evaluar información nueva, y de utilizarla para modificar estimaciones anteriores —basadas en datos limitados— de la probabilidad de que un resultado particular ocurra. Nos permite actuar en base a informaciones parciales

—como a menudo tendrá que hacer un sistema experto— y después evaluar y revisar nuestras decisiones conforme nos llegan más datos.

Para comprender el teorema de Bayes y ver cómo puede valernos para el desarrollo de nuestros propios sistemas expertos, necesitamos saber un poco de probabilidades.

La probabilidad de que un suceso ocurra es el número de casos favorables dividido por el de casos posibles. Esto es, la probabilidad de que una moneda caiga de cara es 1 (número de resultados en los que se da el suceso en cuestión) dividido por 2 (número de resultados posibles, despreciando el caso de que caiga de canto).

Para poner lo anterior en forma de ecuación, en la que P(resultado) indica la probabilidad de tal resultado, podemos escribir:

$$P(\text{resultado}) = \frac{\text{número de resultados favorables}}{\text{número de resultados posibles}}$$

Para el caso de lanzamiento de una moneda, podríamos escribir:

$$P(\text{cara}) = \frac{A}{1+1} = \frac{1}{2} = 0.5$$

Si analizamos más de un suceso (contrariamente al caso del lanzamiento de una moneda en el que solamente nos interesamos en un único lanzamiento de una única moneda), los sucesos pueden ser *mutuamente excluyentes* (si llueve no puede ser que no esté lloviendo) o *no mutuamente excluyentes* (puede hacer frío y haber niebla).

Sucesos mutuamente excluyentes

La probabilidad en el caso de sucesos mutuamente excluyentes es la probabilidad de que el resultado X *o* el resultado Y ocurra. En este caso las probabilidades *se suman*, así:

$$P(\text{resultadoX O resultadoY}) = P(\text{resultadoX}) + P(\text{resultadoY})$$

Supongamos que tiramos un dado. Queremos saber la posibilidad de que salga un tres:

$$P(\text{tres}) = \frac{1}{6}$$

La probabilidad de que salga un cinco, P(cinco), es la misma, un sexto. El dado no puede caer mostrando *dos* números a la vez (no tenemos en cuenta si

está apoyado sobre un objeto), por lo que los sucesos son mutuamente excluyentes. Esto significa que la probabilidad de que un dado caiga mostrando un tres *o* un cinco se puede expresar así:

$$P(\text{tres O cinco}) = P(\text{tres}) + P(\text{cinco}) = \frac{1}{6} + \frac{1}{6} = \frac{2}{6} = \frac{1}{3}$$

En otras palabras, la probabilidad de que al tirar una sola vez un dado salga un tres o un cinco es de un tercio.

Al tirar el lado anterior puede que salga, o no, un tres o un cinco. ¿Cuál es la probabilidad de que NO salga un tres o un cinco? Está claro que esta probabilidad es 2/3 o 1-1/3. La probabilidad de que un suceso *ocurra* más la probabilidad de que *no ocurra* debe ser igual a 1:

$$P(\text{suceda}) + P(\text{no suceda}) = 1$$

...lo que equivale a...

$$P(\text{no suceda}) = 1 - P(\text{suceda})$$

Sucesos que no son mutuamente excluyentes

Supongamos que tenemos una llave que únicamente abre una de las seis cajas que están sobre una mesa enfrente de nosotros y no sabemos qué caja es. Sin embargo, sabemos el contenido de ellas: un cuaderno negro, una bola negra, un cuaderno verde, una bola roja, un cepillo de dientes negro, y un ratón hinchable. Probamos a abrir con la llave cada una de las cajas, hasta que una lo hace. ¿Cuál es la probabilidad de que la caja que abrimos contenga algo negro o una bola? Obviamente estos sucesos no son mutuamente excluyentes, ya que hay un objeto (la bola negra) que satisface ambas condiciones.

No obstante, el abrir una caja que contenga una bola no implica necesariamente que tal bola sea negra. Pero debido a que la posibilidad de abrir una caja cuyo contenido satisfaga ambas condiciones existe, necesitamos reducir la probabilidad de que cada condición quede satisfecha, en la probabilidad de que queden satisfechas ambas.

La probabilidad de obtener un objeto negro es 3/6, y la de obtener una sola 2/6. La ecuación que representa la probabilidad de seleccionar una bola o un objeto negro (es decir, la probabilidad de dos sucesos que no son mutuamente excluyentes) es:

$$P(\text{negro O bola}) = P(\text{negro}) + P(\text{bola}) - P(\text{negro Y bola})$$

En palabras, esta ecuación significa: la probabilidad de escoger un objeto negro o una bola es igual a la probabilidad de escoger un objeto negro más la probabilidad de escoger una bola, menos la probabilidad de escoger un objeto negro que además sea una bola (una bola negra):

$$\begin{aligned} P(\text{negro O bola}) &= 3/6 + 2/6 - 1/6 \\ &= 5/6 - 1/6 \\ &= 4/6 \\ &= 2/3 \end{aligned}$$

Independencia estadística

Tal vez se pregunte qué relación tiene todo esto con el reverendo Bayes. La tiene, y quedará muy clara después. El trabajo de Bayes no se puede explicar hasta que hayamos acabado con las probabilidades.

Si, cuando golpea a alguien en la cara con su poderoso puño-martillo, más tarde se le pone un ojo morado, decimos que los sucesos son estadísticamente dependientes. La probabilidad de que el segundo suceso acaezca (el ojo se va poniendo morado) está íntimamente relacionada con la probabilidad de que usted golpee a alguien en un ojo lo suficientemente fuerte con su puño. Sin embargo, el solo hecho de que los dos sucesos ocurran seguidos no significa que sean siempre estadísticamente dependientes. Lanzamos al aire una moneda y cae de cara. La lanzamos otra vez. La probabilidad de que salga cara en este segundo lanzamiento es totalmente independiente del resultado del anterior lanzamiento.

Si denominamos P(cuatro) a la probabilidad de sacar un cuatro al tirar un dado, y P(seis) a la de sacar un seis, la probabilidad de obtener un cuatro en la primera tirada y un seis en la segunda puede expresarse así:

$$P(\text{cuatro} \cap \text{seis}) = P(\text{cuatro}) \times P(\text{seis})$$

(donde $P(\text{cuatro} \cap \text{seis})$ representa la probabilidad de obtener un cuatro y un seis, juntos o en sucesión).

La probabilidad de obtener un cuatro es 1/6 y la de obtener un seis es también 1/6; entonces, la probabilidad de obtener un cuatro seguido de un seis es 1/6 veces 1/6, es decir, 1/36. La probabilidad de obtener un cuatro, seguido de un seis, seguido de un tres, es 1/6 veces 1/6 veces 1/6 veces, o 1/216. Expresado en forma de ecuación, tenemos:

$$P(\text{cuatro} \cap \text{seis} \cap \text{tres}) = P(\text{cuatro}) \times P(\text{seis}) \times P(\text{tres})$$

La probabilidad de que esto no ocurra (es decir, la probabilidad de que no salga un cuatro en la primera tirada, un seis en la segunda y un tres en la tercera) es uno menos la probabilidad de que ello ocurra:

$$P(\text{NO } (\text{cuatro} \cap \text{seis} \cap \text{tres})) = 1 - P(\text{cuatro}) \times P(\text{seis}) \times P(\text{tres})$$

Este procedimiento de calcular la probabilidad de que un suceso no ocurra es válido para cualquier situación; basta con restar de uno la probabilidad de que el suceso ocurra. Si la probabilidad de obtener un seis al tirar un dado, P(seis), es 1/6, la probabilidad de que no salga un seis es 1 – 1/6, es decir, 5/6.

Probabilidad condicional

La probabilidad condicional se refiere a la posibilidad de que ocurra el suceso Y, dado que *ha* ocurrido el X. Se denota por P(Y/X). Si los sucesos son *estadísticamente independientes*, la probabilidad de Y dado que X ha ocurrido, es (tal vez le sorprenda) simplemente la probabilidad de que ocurra el suceso Y, o sea, P(Y).

¿Por qué es esto así? Si tiramos un dado, la probabilidad de que otra vez salga un seis sigue siendo 1/6. El lanzamiento del dado no influye en los resultados de los lanzamientos siguientes.

Téngase en cuenta que en este caso, de independencia estadística, preguntamos por la probabilidad de que Y ocurre, dado que X *ha* ocurrido y no por la probabilidad de que ocurra el suceso "Y y X".

Dependencia estadística

Lamento tener que decir que las cosas se ponen algo más complicadas cuando la probabilidad de un segundo depende de la probabilidad de otro anterior.

Probabilidad condicional

Imaginemos una situación en la que disponemos de una llave que abre una de las nueve cajas que tenemos delante. Cinco de ellas contienen libros escritos por Tim Hartnell; dos, libros escritos por el Dr. Rodnay Zaks, y las dos restantes, libros de Grace Murray Hopper (diseñadora del lenguaje COBOL).

La probabilidad de que la caja abierta con la llave contenga cualquiera de los libros es 1/9. Tenemos nueve libros y todos ellos tienen la misma probabilidad de ir a parar a la caja que abrirá la llave. Sin embargo, supongamos que la caja que hemos abierto contiene un libro escrito por un varón. La probabilidad de que esto ocurra es 7/9. ¿Cuál es la probabilidad de que el autor sea Rodnay Zaks? Esto lo podemos representar por P(Z/V), la probabilidad de que el libro sea de Zaks, P(Z), dado que el autor del libro es varón, P(V).

Sabemos que el libro es de un autor masculino. Para calcular la probabilidad de que tal autor sea Zaks, ignoramos los libros escritos por Hopper, ya que no vienen a cuento en esta situación. Sabemos también que hay siete libros escritos por autores varones, dos de los cuales son de Zaks. Para hallar las probabilidades, en cuanto a estos siete libros, de que los autores sean Zaks y

Hartnell, respectivamente, dividimos el número de libros de cada uno de ellos, por el número de libros escritos por un varón.

$$P(Z/V) = 2/7$$
$$P(H/V) = 5/7$$

Estas probabilidades añadidas suman uno, tal como cabe esperar. La probabilidad de que la caja abierta contenga un libro escrito por Zaks, dado que el autor es varón, es de 2/7; la probabilidad de que sea de Hartnell, es de 5/7.

Hay una mayor probabilidad, dado que el autor es varón, de que el libro sea de Hartnell a que lo sea de Zaks. Para calcular la probabilidad de que el libro sea de Zaks dado que su autor es varón, P(Z/V), dividimos la probabilidad de Zaks por la probabilidad de que sea un autor varón, P(V), siendo P(V) igual a P(Z) más P(H):

$$P(Z/V) = \frac{P(Z)}{P(V)} = \frac{2/9}{7/9} = \frac{0.2222}{0.7777} = 0.286$$

Como comprobación de lo anterior, podemos razonar que hay siete libros de autores varones. Si el libro que tenemos es de autor varón, hay dos posibilidades (entre siete) de que sea de Zaks. Por tanto, si nuestro método anterior de cálculo de P(Z/V) es correcto, debería dar el mismo resultado, 2/7 (que es la probabilidad de que, entre siete libros, dos de los cuales son de Zaks, el libro elegido sea de él). De hecho así es.

Entonces, la probabilidad condicional, cuando los sucesos son estadísticamente dependientes, se expresa así:

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}$$

donde $P(Y \cap X)$ es la probabilidad de que ocurran juntos los sucesos X e Y.

De vuelta a Bayes

Lo anterior, por fin, nos traslada a una posición desde la que podemos valorar el trabajo del buen reverendo. Como recordará, al principio de esta sección dije que el teorema de Bayes nos proporciona un medio de utilizar información obtenida posteriormente para modificar estimaciones anteriores, basadas en datos limitados, de la probabilidad de que un suceso particular ocurra.

Supongamos que tenemos dos urnas, cada una de ellas con 25 bolas de madera. En una de las urnas (U1), hay 14 negras y 11 rojas. En la otra (U2), hay 19 negras y 6 rojas. Escojamos una urna al azar, metamos la mano y saquemos una bola. Resulta ser negra. ¿Cuál es la probabilidad de que la hayamos sacado de U2?

La probabilidad de seleccionar U1 o U2 es 1/2 (0.5). La probabilidad de sacar una bola negra de U1 es 14/25 (0.56), y la probabilidad de sacarla de U2 es 19/25 (0.76). La probabilidad de sacar una bola negra y precisamente de U1 es 0.5 veces 0.56 (0.28) y la probabilidad de que sea negra y de U2 es 0.5 veces 0.76 (0.38). La probabilidad total de que la bola extraída sea negra es la suma de ambas probabilidades, 0.28 más 0.38 (0.66). La probabilidad, entonces, de que la bola que hemos sacado pertenezca a U1 es (PU1 ∩ negra)/P(negra), o sea, 0.28 dividido por 0.66, que es aproximadamente 0.424, y la probabilidad de que pertenezca a U2 es 1 – 0.424 o 0.576 (como solamente las podemos extraer de U1 y de U2, la suma de ambas probabilidades debe ser igual a 1).

¿Qué nos dice esto? ¿Qué significado tienen el 0.424 o el 0.576? Antes de efectuar la extracción de la bola, habríamos dicho que la probabilidad de que perteneciera a una u otra urna era la misma, 0.5, pero ahora —después de hecha solamente una extracción— podemos decir que es más probable que la bola proceda de U2 que de U1.

Reemplazamos la bola, barajamos las urnas y sacamos otra bola. Supongamos que de nuevo es negra. ¿Podemos decir, con cierta confianza, que las probabilidades anteriores son válidas, después de que ha vuelto a salir de U2? La probabilidad de que las dos bolas negras vinieran de U1 es 0.5 veces 0.56 veces 0.56 (0.159) y la probabilidad de que ambas vinieran de U2 es 0.5 veces 0.76 veces 0.76 (0.289). Si sumamos esto, obtenemos la probabilidad de sacar dos bolas negras de forma consecutiva (0.159 más 0.289 es 0.448).

Ahora, ¿cómo calculamos la probabilidad de que cogiéramos la segunda bola negra de U2?

La probabilidad de sacar dos bolas negras de forma consecutiva de U1 es la probabilidad de escoger U1 y sacar dos bolas negras de ella (0.159) dividida por la probabilidad de sacar dos bolas negras de forma consecutiva (0.448), que es 0.355. La probabilidad de sacar dos bolas negras de forma consecutiva de U2 deberá ser 1 – 0.355, es decir, 0.645. Comprobémoslo: La probabilidad de escoger U2 y sacar dos bolas negras de ella (0.289) dividida por la probabilidad de sacar dos bolas negras de forma consecutiva (0.448), que es 0.645 tal y como lo habíamos previsto.

¿Adónde hemos llegado? ¿Sabemos algo más que al principio? Comenzamos este proceso con la única información de que teníamos una posibilidad entre dos (0.5) de escoger U1 o U2. Después de sacar una sola bola, que resultó ser negra, podíamos decir que la probabilidad de que procediera de U1 era 0.424 y la probabilidad de que viniera de U2 era 0.576. Reemplazamos la bola, barajamos y sacamos otra más. De nuevo resultó ser negra. Hicimos otra vez los cálculos y decidimos que la probabilidad de que la bola viniera de U1 era 0.355 y 0.645 que viniera de U2. Esto nos permite decir que si sacamos dos bolas de forma consecutiva (reemplazando la primera antes de sacar la segunda), y ambas eran negras, la probabilidad de que viniesen de U2 es 0.645.

Apéndice C
Bases de datos

El poder de los sistemas expertos en el futuro se deberá a la eficacia e inteligencia del bloque deductivo, y a la calidad y extensión de la base de conocimientos a la que el sistema pueda acceder.

En el futuro próximo, la información que la base de conocimientos podrá mantener será fundamentalmente de naturaleza textual, ya que los ordenadores actuales son mucho mejores manipulando los símbolos que representan textos que aquellos otros símbolos complejos tales como los que codifican una imagen animada en color. Actualmente, aproximadamente un 55 por 100 del material manejado por un típico ordenador de empresa es texto; el 30 por 100, datos, y el resto, un 15 por 100, información de imágenes.

Al establecer una base de datos para que acceda a ella un sistema experto, hay tres cosas que se deben tener en cuenta. Estas son:

- El coste del mantenimiento de la información.
- La velocidad de comunicación de la información.
- La realidad de la información mantenida.

También se debe tener en cuenta el tamaño total de la información a manipular.

El coste de almacenamiento de información

El coste del mantenimiento de información se ha reducido drásticamente a lo largo de los últimos treinta años. El cuadro siguiente —que muestra los costes brutos del mantenimiento de la información que cabe en un folio— lo ilustra de forma convincente.

Año	*En memoria principal*	*En línea acceso directo*
1950	225 000 000.00 $	—
1960	70 000.00 $	12 000.00 $
1970	11 500.00 $	2 500.00 $
1975	2 750.00 $	250.00 $
1980	275.00 $	20.00 $
1983	120.00 $	15.00 $
1985 (estimación)	12.00 $	0.75 $
1990 (estimación)	2.00 $	0.12 $

Las dos últimas cifras son posiblemente estimaciones muy conservadoras. En 1987, el almacenamiento electrónico será el método más barato disponible para guardar *texto*, y nos ofrecerá, por supuesto, la ventaja adicional de la accesibilidad inmediata, búsqueda veloz y comunicación rápida. Dos años más tarde, el almacenamiento electrónico ofreciéndonos color total, acceso directo y animación, será el medio más barato de guardar información de *imágenes*.

Características de la información en las bases de datos

¿Cuáles son las características de los grandes volúmenes de información, tales como los que se guardarán en las bases de conocimiento? Hay tres niveles. El primero es el de los *datos puros*, un conjunto de hechos inconexos. El segundo nivel es el del conocimiento *estructurado* o categorizado, mantenido de forma que pueda ser manejado. Esta estructura es vital si se quiere que la información tenga algún sentido. El tercero —y presumiblemente el nivel más valioso, en la mayoría de los casos— es el del "conocimiento *asociado*", en el cual no solamente se guarda información, sino que, en él, la base de conocimientos almacena las *relaciones* entre los elementos de la base de datos.

Las decisiones más importantes que se tomarán en la próxima década se referirán a la *forma* en la cual se mantendrán las bases de conocimiento y sus interrelaciones. Hay muchas maneras de almacenar las bases de conocimientos, y la información relativa a las relaciones entre los elementos de esas bases y el

tipo de decisiones que tomemos ahora sobre ellas nos obligará, quizá para siempre, a utilizar tales formas de organización. Un problema actual relacionado con ello es la magnitud total de la tarea de convertir "bases de datos viejas" (mantenidas en formas tales como archivos de papel) en formas electrónicas.

En Gran Bretaña, por ejemplo, el Departamento de Sanidad y Seguridad Social se enfrenta actualmente a dos problemas. Posee actualmente unos 26,5 millones de nombres en archivos activos, relacionados por cosas tales como su localización geográfica, y una enorme "base de reglas" que contiene información tal como la forma de hacer los pagos, a quiénes y bajo qué circunstancias. De acuerdo con este Departamento, la tecnología actual no está capacitada para manejar la "base de reglas". Sin embargo, los métodos actuales no son adecuados por más tiempo, por lo que, incluso antes de que exista la tecnología adecuada, es necesario decidir cómo se mantendrá la información, de forma que, cuando la tecnología esté disponible, no resulten excesivamente perjudicados, al quedarse limitados a formas primitivas. Esto nos indica el tipo de problema con el que se enfrenta gran parte de la actual organización del conocimiento.

Cómo se mantiene la información

En la forma actual de mantener información se distinguen tres componentes principales. El más importante, en el presente, es la base de datos, seguido por el trabajo que ahora se está desarrollando sobre sistemas expertos. El tercer componente, que discutiremos en seguida, está peor definido, pero, aun así, constituye una parte importante de los medios actuales de guardar el conocimiento.

Las bases de datos empezaron como simples matrices bidimensionales, y de ellas han evolucionado hasta lo que ahora se denomina "bases de datos relacionales". El mayor problema de una base de datos es que se necesita saber *de antemano* cómo se organizará la información. Esto, tal como advertiremos si pensamos un poco, puede limitar en gran medida la efectividad y flexibilidad de una base de datos.

El segundo componente de la forma actual de mantener el conocimiento es el desarrollo de bases expertas para manejarlas mediante bloques de inferencia.

El tercer componente puede definirse mejor mediante el término "tecnología de transferencia". Es la capacidad para coger ideas de un área y aplicarlas en otras, o en conjunción con otras tecnologías (tales como la enseñanza, aprendizaje basado en ordenador y la utilización de "video interactivo", donde los discos de laser pueden ser pronto el componente principal).

El disco de laser resuelve limpiamente uno de los actuales problemas y potenciales de las enormes cantidades de información que podemos almacenar ahora. En el momento actual, un disco laser puede dar cabida a 55.000 imágenes fijas; es decir, aproximadamente 700 megabytes de información. Cuestan menos de cinco dólares cada uno en producción masiva. Estas son cifras bastante asombrosas. Se están desarrollando los discos que pueden ser escritos

por el usuario, ya que hasta ahora sólo se pueden leer. Las estadísticas muestran su potencial. El problema radica en el *gobierno* de toda esta información. En contraste con la mayoría de la información textual, las imágenes visuales no son en general tan fácilmente clasificables. No obstante el vasto potencial de referenciación cruzada que posee el ordenador, unido a su capacidad única para manejar datos de la densidad soportada por un disco laser, nos demuestra que los discos video podrían sólo haber evolucionado y merecido la pena cuando se utilizan con un ordenador que controle y organice la información.

La *calidad* de la información guardada en el interior de los bancos de datos de un ordenador hoy día debe ser también analizada. La mayor parte de la información es actualmente manipulada por la lógica matemática (utilizando operadores tales como +, −, *, < y >). Las palabras se tratan como números. Esto significa que —para el ordenador— tienen un valor numérico, pero no significado. Es decir, aunque la información misma pueda manipularse como números, el sistema por sí mismo no puede tener ni la más vaga idea de con qué está trabajando. Si pudiéramos producir un sistema que utilizase las palabras como palabras, manipuladas lógicamente, estaríamos varios pasos más cerca de una máquina "consciente".

Hasta cierto punto, esto ya está ocurriendo. El mayor poder de procesamiento y el inferior coste de almacenaje implican que ahora es posible desarrollar sistemas basados en la lógica verbal (semántica). Estos sistemas reconocen relaciones no matemáticas, como, por ejemplo, pares padre/hijo, grande/extenso y Nueva York/ciudad, tal como ya hemos visto en este libro al comentar los lenguajes HASTE, EASLE, PROLOG-A y SSLISP.

Apéndice D

Reglas de lógica probabilística

La lógica probabilística utiliza los operadores AND, OR y NOT:

NOT: Dadas dos condiciones opuestas, la probabilidad de un estado es (1 – probabilidad) del estado opuesto.

AND: Toma el menor de dos (o más) balores, de modo que si uno es 0.3 y el otro 0.5, al aplicarles el operador AND obtenemos 0.3.

OR: Toma el mayor de dos (o más) valores, de modo que si uno es 0.3 y el otro 0.5, al aplicarles el operador OR obtenemos 0.5.

Observe cómo esta forma de determinar los valores en una situación AND u OR es en gran parte la tradicional. Algunos sostienen que un AND debiera ser el *producto* de la probabilidad-una y la probabilidad-dos. La utilización del método tradicional parece funcionar en la práctica, y dada la forma, en gran parte empírica, según la cual se deben valorar en muchos casos la calidad de las salidas de un sistema como éste, el hecho de que funcione en la práctica es en realidad lo único que importa.

NOT p1 ⟶ 1 – p1
p1 AND p2 ⟶ MIN(p1,p2)
p1 OR p2 ⟶ MAX(p1,p2)

Apéndice E

Datos climatológicos

A continuación vienen los datos utilizados para la predicción del tiempo que efectuamos con FUZZY RITA. Como podrá ver, no todos los valores se utilizaron. Tal vez desee probar usted mismo el sistema, utilizando la información que ignoré en mi prueba.

FECHA	M.S.L. PRESION 9 h. (mb)	TEMPERATURA		H.R. a las 15 h. (%)	VIENTO a las 15 h. KM/H		FUERZA MAX. KM/H		HORA SOLAR	TOTAL EVAP. en 24 H. 9 h. (mm)	TOTAL PRECIP. en 24 H. 9 h. (mm)
		MIN (°C)	MAX (°C)								
1	1011.6	11.0	25.5	31	SE	07	SO	28	13.5	3.4	0
2	1006.7	12.6	27.6	29	SO	11	NO	46	10.0	4.8	0
3	1012.4	10.8	16.5	52	S	13	SO	57	12.7	5.6	2.6
4	1014.7	9.1	15.9	74	O	15	O	72	0.0	4.0	1.6
5	1016.5	11.0	16.8	56	SSE	11	OSO	45	0.7	1.2	7.2
6	1018.1	13.0	17.9	60	S	07	SSO	26	1.2	4.2	0
7	1017.9	12.8	23.1	51	S	13	S	43	11.0	2.4	0
8	1016.7	9.9	22.8	38	SE	19	ESE	50	13.6	3.8	0
9	1013.9	11.6	24.2	38	SE	09	SE	45	13.7	6.0	0
10	1010.8	10.5	27.7	22	E	06	SE	45	11.8	5.6	0
11	999.0	15.9	21.3	60	Calma		NO	50	4.5	6.2	2.6
12	1002.6	12.2	23.5	30	O	26	ONO	63	9.8	3.4	2.6
13	1005.9	13.0	17.9	86	NNO	06	O	30	2.6	4.0	0
14	1000.0	12.5	24.2	84	ENE	04	O	61	3.2	2.4	12.4
15	1003.0	12.5	20.0	46	S	11	OSO	46	10.4	1.2	7.4

FECHA	M.S.L.* PRESION 9 h. (mb)	TEMPERATURA		H.R.** a las 15 h. (%)	VIENTO a las 15 h. KM/H		FUERZA MAX. KM/H		HORA SOLAR	TOTAL EVAP. en 24 H. 9 h. (mm)	TOTAL PRECIP. en 24 H. 9 h. (mm)
		MIN (°C)	MAX (°C)								
16	1009.8	11.4	23.7	43	SE	11	S	41	12.9	5.8	0.4
17	1011.1	12.1	27.0	31	SSE	07	N	26	13.5	3.2	0
18	1002.0	16.8	24.6	36	OSO	26	OSO	67	10.0	6.8	0.2
19	1003.7	12.1	25.0	38	NO	19	O	67	7.6	7.0	0
20	1000.1	11.3	20.0	40	OSO	26	O	72	11.1	4.4	0.8
21	1010.1	10.4	22.7	37	O	11	OSO	43	9.1	2.8	1.2
22	1013.6	14.2	22.6	40	SSE	07	OSO	35	8.7	5.4	0
23	1011.8	11.4	29.2	32	SSE	06	S	43	13.4	5.6	0
24	1009 8	13.7	27.8	58	S	06	SSO	52	3.7	5.4	0
25	994.2	16.5	26.5	56	OSO	19	OSO	59	2.0	5.0	1.8
26	1012.0	12.8	18.9	66	SO	19	S	52	4.3	3.0	3.2
27	1017.6	12.2	23.2	41	ESE	13	E	41	14.1	6.6	2.0
28	1017.5	12.0	24.2	40	SE	11	ESE	48	13.4	5.4	0
29	1014.4	12.6	28.1	46	E	11	E	35	12.8	4.6	0
30	1009.6	16.0	30.3	37	S	15	S	41	13.1	5.6	0
31	1007.1	16.4	25.6	46	E	04	S	52	9.2	7.2	0
MEDIA	1009.5	12.6	23.4	47					9.2	142.0	46.0
MEDIA A LARGO PLAZO	1013.2	12.7	24.1	49					8.2	195.3	57.2

* M.S.L. = Satélite meteorológico *(Meteorological Satellite Laboratory).*
** H.R. = Humedad relativa.

Apéndice F

Referencias para consulta

Clark, K. L., y McCabe, F. G., *Micro-PROLOG: Programming in Logic*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

Buchanan, B., y Fiegenbaum, E., "Dendral and Meta-Dendral: Their Applications Dimension", en *Reading in Artificial Intelligence*, B. L. Webber y N. K. Nilsson (eds.), Tioga Publishing, Palo Alto, California, 1981.

Buchanan, B., y Shortliffe, E., *Rule-Based Expert Systems*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

Duda, R.; Gaschni, J., y Hart, P., "Model Design in the PROSPECTOR Consultant System for Mineral Exploration", en *Readings in Artificial Intelligence*, B. L. Webber y N. J. Nilsson (eds.), Tioga Publishing, Palo Alto, California, 1981.

Fiegenbaum, E. A., y McCorduck, P., *The Fifth Generation*, Addison-Wesley Publishing Company, Reading Massachusetts, 1983.

Hartnell, T., *Inteligencia Artificial: conceptos y programas*, Anaya Multimedia, 1985.

McCorduck, P., *Machines Who Think*, W. H. Freeman & Company, San Francisco, 1976.

Raphael, B., *The Thinking Computer*, W. H. Freeman & Company, San Francisco, 1976.

Rich, E., *Artificial Intelligence*, McGraw-Hill, Nueva York, 1983.

Shortliffe, E., "Consultation Systems for Physicians", en *Readings in Artificial Intelligence*, B. L. Webber y N. J. Nilsson (eds.), Tioga Publishing, Palo Alto, California, 1981.

Webber, B. L., y Nilsson, N. J., *Readings in Artificial Intelligence*, Tioga Publishing, Palo Alto, California, 1981.
White, D., y Shaw, W. P., *A Modern Introduction to Chemistry*, Pergamon Press, Elsmford, Nueva York, 1980.
Zadeh, L. A., "A Theory of Approximate Reasoning", en *Machine Intelligence 9*, Hayes, J. E.; D. Michie, y L. I. Mikulich (eds.), Halsted Press, John Wiley & Sons, Nueva York, 1979.

Apéndice G
Otras lecturas

• Sistemas expertos e inteligencia artificial

Boden, M., *Artificial Intelligence and Naturan|Man*, Harvester Press, Basic Books Inc., Nueva York, 1981.

James, M., *Artificial Intelligence in BASIC*, Newnes Technical Books, Butterworth & Company, Londres, 1984.

Naylor, C. M., *Build your own Expert System*, Sigma Technical Press, Cheshire, 1983.

Pearl, J., *Heuristics —Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

Torrance, S. (ed.), *The Mind and the Machine*, Halsted Press, John Wiley & Sons, Nueva York, 1984.

Simons, G. L., *Towards Fith-Generation Computers*, NCC Publications, Manchester, 1983.

Simons, G. L., *Introducing Artificial Intelligence*, NCC Publications, Manchester, 1984.

Winston, P. H., *Artificial Intelligence*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

Wos, L.; Overbeek, R.; Lusk, E., y Boyle, J., *Automated Reasoning: Introduction and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

- **Creación de compiladores**

Nicholls, J. E., *The Structure and Design of Programming Languages*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1975.
Rohl, J. S., *An Introduction to Compiler Writing*, American Elsevier, Nueva York, 1975.

- **LISP**

Berk, A. A., *LISP. El lenguaje de la inteligencia artificial*, Anaya Multimedia, Madrid, 1986.
Siklossy, L., *Let's Talk LISP*, Prentice-Hall Englewood Cliffs, New Jersey, 1976.
Editors (Gnosis), *Learning LISP*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

- **PROLOG**

Berk, A. A., *PROLOG. Programación y aplicaciones en Inteligencia Artificial*, Anaya Multimedia, Madrid, 1986.
McCabe, J. G.; Clark, K. L., y Steel, B. D., *Micro-PROLOG 3.1 Programmer's Reference Manual*, Logic Programming Associates, Londres, 1984.
Pountain, D., "Prolog on Microcomputers", en *Byte Magazine*, McGraw-Hill, diciembre, 1984.

Apéndice H

Adaptaciones para Spectrum, Amstrad, Commodore 64 y Apple II

Programas para Spectrum

En la versión para Spectrum del programa SSLISP se han utilizado las siguientes órdenes:

MSX	*SPECTRUM*	*MSX*	*SPECTRUM*	*MSX*	*SPECTRUM*
CAR	PRIMERO	LISTA	LISTA	MENOS	MENOS
CDR	SEGUNDO	INC1	INC1	RESTA	DIFERENCIA
CONS	CONDENSA	DEC1	DEC1	EXP	EXPT
ATOM	ELEMENTO	CERO?	CERO?	RECIP	INVERSO
IGUAL	IGUAL	UNO?	UNO?	DIVIDE	COCIENTE
NULO	NULO	NUMERO?	NUMERO?	RESTO	RESTO
MIEMBRO	MIEMBRO	NEGATIVO?	NEGATIVO?	SUMA	SUMA
MIGUAL	MIEMIGUAL	MAYOR	MAYOR	MULT	PRODUCTO
JUNTA	JUNTA	MENOR	MENOR	DEFINE	DEFINE
INVIERTE	INVIERTE	MAX	MAX		
MISMO	MISMO	MIN	MIN		

SSLISP

```
  10 REM **********************
  20 REM *                    *
  30 REM *      S.S.LISP      *
  40 REM *                    *
  50 REM **********************
  60 REM
  70 GO TO 3500: REM INICIO
  80 REM ***********************
  90 LET A$=A$(E TO LEN A$-1)
 100 RETURN
 110 REM ***********************
 120 PRINT
 130 LET NN=0
 140 INPUT  LINE A$: PRINT "> ";
A$
 150 IF A$="" THEN  BEEP 1,-15:
STOP : REM PARA FINALIZAR BASTA
 CON PULSAR <ENTER>
 160 REM ***********************
 170 FOR J=1 TO 12
 180 LET X(J)=0: LET Y(J)=0: LET
 Z(J)=0
 190 NEXT J
 200 REM ***********************
 210 LET R=0: LET S=0: LET T=0:
LET CUNO=0: LET CDOS=0: LET ED=0
 220 FOR J=1 TO LEN A$
 230 LET B$=A$(J)
 240 IF B$="(" THEN  LET S=S+1:
LET Z(S)=J: IF T=0 THEN  LET CUN
O=J
 250 IF B$=")" THEN  LET T=T+1:
LET Y(T)=J: IF CDOS<>0 AND ED=0
THEN  LET ED=J-1
 260 IF T=1 AND B$=")" THEN  LET
 CDOS=J
 270 IF B$=" " THEN  LET R=R+1:
LET X(R)=J
 280 NEXT J
 290 IF S=T THEN  GO TO 350: REM
 () EQUILIBRADOS
 300 IF S<T THEN  PRINT " -> OMI
SION DE ("
 310 IF S>T THEN  PRINT " -> OMI
SION DE )"
 320 INPUT "  +   "; LINE B$
 330 LET A$=A$+B$
 340 GO TO 170
 350 IF NWDS=0 OR NN=1 THEN  GO
TO 450
 360 LET M$=A$( TO X(1)-1)
 370 LET FIN=10
 380 FOR J=1 TO NWDS
 390 LET H$=N$(J): GO SUB 3580:
LET W$=H$
 400 LET H$=O$(J): GO SUB 3580
 410 IF M$=W$ THEN  LET A$=H$+A$
(LEN (W$)+1 TO )
 420 NEXT J
 430 LET NN=1
 440 GO TO 170
 450 LET FLAG=0: LET B$="FALSO"
 455 IF LEN A$<5 THEN  GO TO 120
 460 IF A$( TO 5)="MAX (" THEN
LET FLAG=19: GO SUB 2640: GO TO
770
 470 IF A$( TO 5)="MIN (" THEN
LET FLAG=20: GO SUB 2830: GO TO
770
 475 IF LEN A$<6 THEN  GO TO 120
 480 IF A$( TO 6)="NULO (" THEN
 LET FLAG=6: GO SUB 1340: GO TO
770
 490 IF A$( TO 6)="INC1 (" THEN
 LET FLAG=14: GO SUB 2220: GO TO
 770
 500 IF A$( TO 6)="DEC1 (" THEN
 LET FLAG=15: GO SUB 2270: GO TO
 770
 510 IF A$( TO 6)="EXPT (" THEN
 LET FLAG=18: GO SUB 2600: GO TO
 770
 520 IF A$( TO 6)="SUMA (" THEN
 LET FLAG=21: GO SUB 2860: GO TO
 770
 530 IF A$( TO 6)="UNO? (" THEN
 LET FLAG=31: GO SUB 2320: GO TO
 770
 535 IF LEN A$<7 THEN  GO TO 120
 540 IF A$( TO 7)="IGUAL (" THEN
  LET FLAG=5: GO SUB 1210: GO TO
 770
 550 IF A$( TO 7)="LISTA (" THEN
  LET FLAG=12: GO SUB 2170: GO T
O 770
 560 IF A$( TO 7)="CERO? (" THEN
  LET FLAG=16: GO SUB 2320: GO T
O 770
 570 IF A$( TO 7)="MISMO (" THEN
  LET FLAG=11: GO SUB 2040: GO T
O 770
 580 IF A$( TO 7)="MENOS (" THEN
  LET FLAG=22: GO SUB 2910: GO T
O 770
 590 IF A$( TO 7)="MENOR (" THEN
  LET FLAG=28: GO SUB 3200: GO T
O 770
 600 IF A$( TO 7)="JUNTA (" THEN
```

```
 LET FLAG=9: GO SUB 1710: GO TO
 770
 610 IF A$( TO 7)="MAYOR (" THEN
 LET FLAG=27: GO SUB 3160: GO T
O 770
 620 IF A$( TO 7)="RESTO (" THEN
 LET FLAG=25: GO SUB 3070: GO T
O 770
 625 IF LEN A$<8 THEN  GO TO 120
 630 IF A$( TO 8)="DEFINE (" THE
N  LET FLAG=13: GO SUB 3340: GO
TO 770
 635 IF LEN A$<9 THEN  GO TO 120
 640 IF A$( TO 9)="PRIMERO (" TH
EN  LET FLAG=1: GO SUB 850: GO T
O 770
 650 IF A$( TO 9)="SEGUNDO (" TH
EN  LET FLAG=2: GO SUB 900: GO T
O 770
 660 IF A$( TO 9)="INVERSO (" TH
EN  LET FLAG=24: GO SUB 3000: GO
 TO 770
 670 IF A$( TO 9)="MIEMBRO (" TH
EN  LET FLAG=7: GO SUB 1390: GO
TO 770
 680 IF A$( TO 9)="NUMERO? (" TH
EN  LET FLAG=30: GO SUB 3290: GO
 TO 770
 685 IF LEN A$<10 THEN  GO TO 12
0
 690 IF A$( TO 10)="CONDENSA ("
THEN  LET FLAG=3: GO SUB 980: GO
 TO 770
 700 IF A$( TO 10)="ELEMENTO ("
THEN  LET FLAG=4: GO SUB 1120: G
O TO 770
 710 IF A$( TO 10)="PRODUCTO ("
THEN  LET FLAG=26: GO SUB 3110:
GO TO 770
 720 IF A$( TO 10)="INVIERTE ("
THEN  LET FLAG=10: GO SUB 1770:
GO TO 770
 730 IF A$( TO 10)="COCIENTE ("
THEN  LET FLAG=23: GO SUB 2960:
GO TO 770
 735 IF LEN A$<11 THEN  GO TO 12
0
 740 IF A$( TO 11)="MIEMIGUAL ("
 THEN  LET FLAG=8: GO SUB 1520:
GO TO 770
 750 IF A$( TO 11)="NEGATIVO? ("
 THEN  LET FLAG=29: GO SUB 3240:
 GO TO 770
 755 IF LEN A$<12 THEN  GO TO 12
0
 760 IF A$( TO 12)="DIFERENCIA (
" THEN  LET FLAG=17: GO SUB 2390
 770 IF FLAG<>0 THEN  GO SUB 790
 780 GO TO 120
 790 REM ** CONTESTACION **
 800 PRINT "   SU VALOR ES..."
 810 IF B$<>"()" THEN  PRINT "
 ";B$
 820 IF B$="()" THEN  PRINT "
NADA"
 830 RETURN
 840 REM ***********************
 850 REM     **  PRIMERO  **
 860 IF S=2 THEN  LET B$=A$(Z(2)
+1 TO X(2))
 870 IF S>2 THEN  LET B$=A$(CUNO
 TO CDOS)
 880 RETURN
 890 REM ***********************
 900 REM     **  SEGUNDO   **
 910 GO SUB 850
 920 LET LB=LEN B$+11
 930 LET B$="("+A$(LB TO ED+1)
 940 IF B$(LEN B$-1 TO LEN B$)="
))" THEN  LET B$=B$( TO LEN B$-1
)
 950 IF B$(2)=" " THEN  LET B$="
("+B$(3 TO )
 960 RETURN
 970 REM ***********************
 980 REM     **  CONDENSA   **
 990 LET B$=A$(11 TO LEN A$-1)
1000 LET J=0
1010 IF B$(1)="(" THEN  LET J=1
1020 LET J=J+1
1030 IF B$(J)="(" THEN  GO TO 10
60
1040 IF J<LEN B$ THEN  GO TO 102
0
1050 LET B$="** ERROR EN PRIMERO
 **": RETURN
1060 LET LB=LEN B$-1
1070 LET B$="("+B$( TO J-1)+B$(J
+1 TO )
1080 LET B$=B$( TO LB+1)
1090 IF B$(LEN B$-1 TO )=" )" TH
EN  LET B$=B$( TO LEN B$-2)+")"
1100 RETURN
1110 REM ***********************
1120 REM     **  ELEMENTO  **
1130 LET A$=A$(11 TO LEN A$-1)
1140 LET J=0
1150 LET J=J+1
1160 IF A$(J)=" " OR A$(J)="(" T
HEN  RETURN
1170 IF J<LEN A$ THEN  GO TO 115
0
1180 LET B$="CIERTO"
1190 RETURN
1200 REM ***********************
1210 REM      **  IGUAL  **
```

```
1220 LET E=8: GO SUB 90
1230 LET J=0
1240 LET J=J+1
1250 IF A$(J)=")" THEN  RETURN
1260 IF A$(J)=" " THEN  GO TO 12
90
1270 IF J<LEN A$ THEN  GO TO 124
0
1280 RETURN
1290 LET C$=A$( TO J-1)
1300 LET A$=A$(J+1 TO )
1310 IF C$=A$ THEN  LET B$="CIER
TO"
1320 RETURN
1330 REM ***********************
1340 REM      **  NULO  **
1350 IF A$="NULO ()" THEN  LET B
$="**  UTILIZACION ILEGAL  **
   NULO REQUIERE UN ARGUMENTO"
1360 IF A$="NULO (())" THEN  LET
 B$="CIERTO"
1370 RETURN
1380 REM ***********************
1390 REM      **  MIEMBRO  **
1400 LET C$=A$(10 TO )
1410 LET J=1
1420 LET J=J+1
1430 IF C$(J)=")" OR C$(J)="(" T
HEN  LET D$=C$( TO J): GO TO 146
0
1440 IF J<LEN C$ THEN  GO TO 142
0
1450 RETURN
1460 LET J=LEN D$
1470 LET J=J+1
1480 IF C$(J TO J+LEN D$-1)=D$ T
HEN  LET C$=C$( TO LEN C$-1): GO
 TO 1640
1490 IF J<LEN C$-LEN D$ THEN  GO
 TO 1470
1500 RETURN
1510 REM ***********************
1520 REM   **  MIEMIGUAL  **
1530 LET C$=A$(12 TO )
1540 LET J=0
1550 LET J=J+1
1560 IF C$(J)=" " THEN  GO TO 15
90
1570 IF J<LEN A$ THEN  GO TO 155
0
1580 RETURN
1590 LET D$=C$( TO J)
1600 LET C$=C$(J+2 TO )
1610 LET C$=C$( TO LEN C$-2)+" "
1620 LET J=0
1630 LET J=J+1
1640 IF C$(J TO J+LEN D$-1)=D$ T
HEN  LET B$="("+C$(J TO ): GO TO
 1670
1650 IF J<LEN C$-LEN D$ THEN  GO
 TO 1630
1660 RETURN
1670 LET B$=B$( TO LEN B$-1)+")"
1680 IF B$(LEN B$-2 TO )=")))" T
HEN  LET B$=B$( TO LEN B$-1): GO
 TO 1680
1690 RETURN
1700 REM ***********************
1710 REM     **  JUNTA  **
1720 LET B$=A$(8 TO )
1730 LET B$=B$( TO Y(1)-8)+" "+B
$(Z(3)-6 TO )
1740 LET B$=B$( TO LEN B$-1)
1750 RETURN
1760 REM ***********************
1770 REM     **  INVIERTE  **
1780 LET B$=""
1790 LET A$=A$(12 TO ): LET A$=A
$( TO LEN A$-2)
1800 LET CT=0
1810 LET J=0
1820 LET J=J+1: IF J>LEN A$ THEN
  GO TO 1930
1830 IF A$(J)=" " THEN  GO TO 18
60
1840 IF A$(J)="(" THEN  GO TO 18
70
1850 GO TO 1820
1860 LET CT=CT+1: LET G$(CT)=A$(
 TO J-1): LET A$=A$(J+1 TO ): GO
 TO 1810
1870 LET J=J+1: IF A$(J TO J+1)=
"))" THEN  GO TO 2020
1880 IF A$(J)=")" THEN  GO TO 19
20
1890 IF J=LEN A$ THEN  GO TO 191
0
1900 GO TO 1870
1910 LET CT=CT+1: LET G$(CT)=A$+
")": GO TO 1940
1920 LET CT=CT+1: LET G$(CT)=A$(
 TO J): LET A$=A$(J+1 TO ): GO T
O 1810
1930 LET CT=CT+1: LET G$(CT)=A$
1940 FOR M=CT TO 1 STEP -1
1950 LET H$=G$(M): LET FIN=30
1960 GO SUB 3580
1970 IF H$(1)=" " THEN  LET H$="
"
1980 LET B$=B$+H$: IF M>1 THEN
LET B$=B$+" "
1990 NEXT M
2000 LET B$="("+B$+")"
2010 RETURN
2020 LET CT=CT+1: LET G$(CT)=A$(
 TO J+1): LET A$=A$(J+2 TO ): GO
```

```
 TO 1810
2030 REM ***********************
2040 REM       **  MISMO  **
2050 LET E=8: GO SUB 90
2060 LET M=CODE A$: IF M>47 AND
M<58 THEN  GO TO 2410
2070 LET J=0
2080 LET J=J+1
2090 IF A$(J TO J+1)=") " THEN
LET J=J+1: GO TO 1290
2100 IF A$(J TO J+2)=")))" THEN
 GO TO 2140
2110 IF A$(J TO J+1)="))" THEN
GO TO 2150
2120 IF J<LEN A$ THEN  GO TO 208
0
2130 RETURN
2140 LET C$=A$( TO J+2): LET A$=
A$(J+4 TO ): GO TO 1310
2150 LET C$=A$( TO J+1): LET A$=
A$(J+3 TO ): GO TO 1310
2160 REM ***********************
2170 REM       **  LISTA  **
2180 LET E=8: GO SUB 90
2190 LET B$="("+A$+")"
2200 RETURN
2210 REM ***********************
2220 REM       **  INC1  **
2230 LET E=7: GO SUB 90
2240 LET B$=STR$ (VAL A$+1)
2250 RETURN
2260 REM ***********************
2270 REM       **  DEC1  **
2280 LET E=7: GO SUB 90
2290 LET B$=STR$ (VAL A$-1)
2300 RETURN
2310 REM ***********************
2320 REM       **  CERO?  **
2330 IF FLAG=16 THEN  LET E=8
2340 IF FLAG=31 THEN  LET E=7
2350 GO SUB 90
2360 IF A$="0" AND FLAG=16 OR A$
="1" AND FLAG=31 THEN  LET B$="C
ORRECTO"
2370 RETURN
2380 REM ***********************
2390 REM ** DOS ARGUMENTOS  **
2400 LET E=13: GO SUB 90
2410 LET J=0
2420 LET J=J+1
2430 IF A$(J)=" " THEN  GO TO 24
60
2440 IF J<LEN A$ THEN  GO TO 242
0
2450 LET B$="          ** ERROR
**            SE NECESITAN DOS AR
GUMENTOS": RETURN
2460 LET P=VAL A$( TO J-1)
2470 LET Q=VAL A$(J+1 TO )
2480 IF FLAG=17 THEN  LET B$=STR
$ (P-Q): RETURN
2490 IF FLAG=23 OR FLAG=25 THEN
 LET B=P/Q
2500 IF FLAG=25 THEN  LET B=(INT
 (.5+Q*(B-INT B)*1000))/1000
2510 IF FLAG=18 THEN  LET B=P^Q
2520 IF FLAG=11 AND P=Q THEN  LE
T B$="CORRECTO"
2530 IF FLAG=27 AND P>Q THEN  LE
T B$="CORRECTO"
2540 IF FLAG=28 AND P<Q THEN  LE
T B$="CORRECTO"
2550 IF FLAG=32 THEN  LET B=P-Q
2560 IF FLAG=11 OR FLAG>26 THEN
 RETURN
2570 LET B$=STR$ (B)
2580 RETURN
2590 REM ***********************
2600 REM       **  EXPT  **
2610 LET E=7: GO SUB 90
2620 GO TO 2410
2630 REM ***********************
2640 REM       **  MAX  **
    ** (MIN SUMA PRODUCTO) **
2650 LET F$=A$( TO 3): LET A$=A$
(6 TO )
2660 LET CT=0: LET FLAG=0
2670 IF F$="PRODUCTO" THEN  LET
CT=1
2680 LET J=0
2690 LET J=J+1
2700 IF A$(J)=" " THEN  GO TO 27
30
2710 IF J<LEN A$ THEN  GO TO 269
0
2720 IF J=LEN A$ THEN  LET FLAG=
1
2730 LET P=VAL A$( TO J-1): IF F
LAG=0 THEN  LET A$=A$(J+1 TO )
2740 IF F$<>"SUMA" AND CT=0 THEN
  LET CT=P
2750 IF F$="MAX" AND P>CT THEN
LET CT=P
2760 IF F$="MIN" AND P<CT THEN
LET CT=P
2770 IF F$="SUMA" THEN  LET CT=C
T+P
2780 IF F$="PRODUCTO" THEN  LET
CT=CT*P
2790 IF FLAG=0 THEN  GO TO 2680
2800 LET B$=STR$ (CT)
2810 RETURN
2820 REM ***********************
2830 REM       **  MIN  **
2840 GO TO 2650
2850 REM ***********************
```

```
2860 REM        **  SUMA  **
2870 LET F$="SUMA"
2880 LET A$=A$(7 TO )
2890 GO TO 2660
2900 REM ***********************
2910 REM        **  MENOS  **
2920 LET E=8: GO SUB 90
2930 LET B$=STR$ (-VAL A$)
2940 RETURN
2950 REM ***********************
2960 REM    **  COCIENTE **
2970 LET E=11: GO SUB 90
2980 GO TO 2410
2990 REM ***********************
3000 REM    **  INVERSO  **
3010 LET E=10: GO SUB 90
3020 IF A$="0" THEN  LET B$="* E
RROR:DIVISION POR CERO* ": RETURN
3030 LET B=1/(VAL A$)
3040 LET B$=STR$ B
3050 RETURN
3060 REM ***********************
3070 REM        **  RESTO **
3080 LET E=8: GO SUB 90
3090 GO TO 2410
3100 REM ***********************
3110 REM    **  PRODUCTO  **
3120 LET F$="PRODUCTO"
3130 LET A$=A$(11 TO )
3140 GO TO 2660
3150 REM ***********************
3160 REM        **  MAYOR  **
3170 LET E=8: GO SUB 90
3180 GO TO 2410
3190 REM ***********************
3200 REM        **  MENOR  **
3210 LET E=8: GO SUB 90
3220 GO TO 2410
3230 REM ***********************
3240 REM    **  NEGATIVO?  **
3250 LET E=12: GO SUB 90
3260 IF VAL A$<0 THEN  LET B$="C
IERTO"
3270 RETURN
3280 REM ***********************
3290 REM     **  NUMERO?  **
3300 LET A$=A$(10 TO )
3310 IF CODE A$>44 AND CODE A$<5
8 THEN  LET B$="CIERTO"
3320 RETURN
3330 REM ***********************
3340 REM     **  DEFINE  **
3350 LET A$=A$(9 TO )
3360 LET F$=A$( TO X(2)-9)
3370 LET L$=A$(X(4)-6 TO )
3380 LET J=0
3390 LET J=J+1
3400 IF L$(J)=" " THEN  GO TO 34
30
3410 IF J<LEN L$ THEN  GO TO 339
0
3420 LET B$="* ERROR:DEFINE INCO
RRECTO *": RETURN
3430 LET L$=L$( TO J-1)
3440 LET NWDS=NWDS+1
3450 LET O$(NWDS)=L$: LET N$(NWD
S)=F$
3460 LET B$=F$
3470 RETURN
3480 REM ***********************
3490 REM *** INICIALIZACION  ***
3500 REM ***********************
3510 POKE 23658,8
3520 POKE 23609,40
3530 POKE 23692,255
3540 BORDER 1: PAPER 1: INK 7: C
LS
3550 DIM G$(20,30): DIM O$(20,10
): DIM N$(20,10): DIM X(12): DIM
 Y(12): DIM Z(12)
3560 LET NWDS=0: REM CONTADOR DE
 FUNCIONES DEFINIDAS POR EL USUA
RIO
3570 GO TO 120
3580 REM ELIMINACION BLANCOS
3590 FOR W=FIN TO 1 STEP -1
3600 IF H$(W)<>" " THEN  LET H$=
H$( TO W): GO TO 3620
3610 NEXT W
3620 RETURN
```

PROLOG-A

```
  1 REM ***********************
  2 REM *                     *
  3 REM *      PROLOG-A       *
  4 REM *                     *
  5 REM ***********************
  6 REM
 10 PAPER 1: BORDER 1: INK 7
 15 POKE 23609,40: POKE 23658,8
: POKE 23692,255
 20 REM * TODAS LAS ENTRADAS
        EN MAYUSCULAS *
 30 GO TO 50
```

```
  40 PRINT "NO HAY (MAS) RESPUES
TAS": RETURN
  50 GO SUB 3270: REM INICIO
  60 REM ************************
  70 PRINT
  75 PRINT AT 0,0;"
                                "
  80 INPUT "&.": LINE J$
  90 IF J$="" THEN  STOP
  92 CLS
  95 PRINT  INK 7; PAPER 0; FLAS
H 1;AT 0,8;"ESPERE,POR FAVOR"
 100 IF J$="LISTA TODO" THEN  GO
 SUB 1860: GO TO 70
 110 IF J$( TO 6)="LISTA " THEN
 LET J$=J$(6 TO )+" ": GO SUB 99
0: GO TO 70
 120 IF J$(LEN J$ TO )<>")" THEN
  PRINT "1.": INPUT L$: LET J$=J
$+L$: GO TO 120
 130 LET LJ=LEN J$
 140 LET J$=J$( TO LJ-1)+" ":
 REM SUSTITUCION DEL ULTIMO ) PO
RUN ESPACIO
 150 LET LJ=LEN J$
 160 LET FL=0
 170 IF J$( TO 5)="METE(" THEN
LET J$=J$(6 TO ): LET FL=1
 180 LET RU=0: LET MA=0: LET AR=
0
 190 FOR R=1 TO LEN J$
 195 IF (R+3)>LEN J$ THEN  GO TO
 205
 200 IF J$(R TO R+3)=" SI " THEN
  LET RU=R: LET FL=6
 205 IF (R+2)>LEN J$ THEN  GO TO
 215
 210 IF J$(R TO R+2)=" Y " THEN
 LET MA=R
 215 IF (R+4)>LEN J$ THEN  GO TO
 225
 220 IF J$(R TO R+4)="SUMA(" THE
N  LET AR=1
 225 IF (R+5)>LEN J$ THEN  GO TO
 235
 230 IF J$(R TO R+5)="VECES(" TH
EN  LET AR=2
 235 IF (R+6)>LEN J$ THEN  GO TO
 245
 240 IF J$(R TO R+6)=" MENOR " T
HEN  LET AR=3
 245 IF (R+2)>LEN J$ THEN  GO TO
 260
 250 IF J$(R TO R+2)="ENT" THEN
 LET AR=4
 260 NEXT R
 265 IF LEN J$<3 THEN  GO TO 275
 270 IF J$( TO 3)="ES(" THEN  LE
```

```
T J$=J$(4 TO ): LET FL=2
 275 IF LEN J$<9 THEN  GO TO 285
 280 IF J$( TO 9)="CUAL(X : " TH
EN  LET J$=J$(10 TO ): LET FL=3
 285 IF LEN J$<15 THEN  GO TO 30
0
 290 IF J$( TO 15)="CUAL((X Z) :
 X " THEN  LET J$=J$(16 TO ): LE
T FL=4
 300 IF FL=0 THEN  PRINT "ERROR
DE SINTAXIS": GO TO 70
 310 LET LJ=LEN J$
 320 REM AHORA BUSCA LAS
    SUBRUTINAS PERTINENTES
 330 IF MA<>0 THEN  GO SUB 1950:
 GO TO 70: REM
   CODIFICA LA REGLA "Y"
 340 IF RU<>0 AND FL<>5 THEN  GO
 SUB 1110: REM
   CODIFICACION DE LA REGLA
 350 IF AR<>0 THEN  GO SUB 2430:
 GO TO 70: REM ARITMETICA
 360 IF J$(LEN J$-2 TO )=" X " O
R J$(LEN J$-2 TO )=" Z " THEN  L
ET J$=J$( TO LJ-2)+" "
 370 LET LJ=LEN J$
 380 IF FL=1 THEN  GO SUB 440:
  REM METE
 390 IF FL=2 THEN  GO SUB 520:
  REM ES
 400 IF FL=3 THEN  GO SUB 610:
  REM CUAL
 410 IF FL=4 THEN  GO SUB 830:
  REM CUAL2
 420 GO TO 70
 430 REM ************************
 440 REM             METE
      (ADQUIERE INFORMACIONES)
 450 LET K=0
 460 LET K=K+1
 470 IF CODE Z$(K)=32 THEN  LET
Z$(K)=J$+"@": BEEP .1,10: BEEP .
3,15: RETURN
 480 IF K<500 THEN  GO TO 460
 490 PRINT  INK 5; FLASH 1;"MEMO
RIA LLENA": BEEP .5,15
 500 RETURN
 510 REM ************************
 520 REM               ES
 530 LET K=0
 540 LET K=K+1
 550 IF CODE Z$(K)=32 THEN  GO T
O 580
 560 LET A$=Z$(K): GO SUB 8500:
IF A$=J$ THEN  PRINT  INK 6;"SI"
: GO TO 590
 570 IF K<500 THEN  GO TO 540
 580 PRINT  INK 6;"NO"
```

```
 590 BEEP .1,10: BEEP .3,15: RET
URN
 600 REM ***********************
 610 REM           CUAL
   (BUSCA LAS CORRESPONDENCIAS)
 620 IF J$(1)="X" THEN  GO TO 71
0
 630 LET J$=J$( TO LJ-1)
 640 LET K=0
 650 LET K=K+1
 660 IF CODE Z$(K)=32 THEN  GO T
O 690
 670 LET A$=Z$(K): GO SUB 8500:
IF J$=A$( TO LEN J$) THEN  PRINT
  INK 6;A$(LEN J$ TO )
 680 IF K<500 THEN  GO TO 650
 690 GO SUB 40
 695 BEEP .1,10: BEEP .3,15
 700 RETURN
 710 REM PREGUNTA INICIADA CON X
 720 LET J$=J$(3 TO LEN J$)
 730 LET LJ=LEN J$
 740 LET K=0
 750 LET K=K+1
 760 IF CODE Z$(K)=32 THEN  GO T
O 800
 770 LET A$=Z$(K): GO SUB 8500:
LET Q$=A$(LEN A$-LJ+1 TO )
 780 IF Q$=J$ THEN  PRINT  INK 6
;A$( TO LEN A$-LJ-1)
 790 IF K<500 THEN  GO TO 750
 800 GO SUB 40
 805 BEEP .1,10: BEEP .3,15
 810 RETURN
 820 REM ***********************
 830 REM           CUAL2
 840 LET J$=J$( TO LJ-2)
 850 LET LJ=LEN J$
 860 LET K=0
 870 LET K=K+1
 880 IF CODE Z$(K)=32 THEN  GO T
O 960
 890 LET LF=0
 895 LET A$=Z$(K)
 900 FOR L=1 TO 30-LJ
 910 IF A$(L TO L+LJ-1)=J$ THEN
 LET LF=L
 920 NEXT L
 930 IF LF=0 THEN  GO TO 950
 935 GO SUB 8500
 940 PRINT A$( TO LF-2);A$(LF+LJ
 TO )
 945 BEEP .05,0
 950 IF K<500 THEN  GO TO 870
 960 GO SUB 40
 965 BEEP .1,10: BEEP .3,15
 970 RETURN
 980 REM ***********************
 990 REM           LISTA
1000 LET K=0
1010 LET K=K+1
1020 IF CODE Z$(K)=32 THEN  RETU
RN
1030 LET LF=0
1040 FOR L=1 TO LEN Z$(K)-LEN J$
1050 LET A$=Z$(K): IF A$(L TO L+
LEN J$-1)=J$ THEN  LET LF=1
1060 NEXT L
1070 IF LF=1 THEN  LET A$=Z$(K):
 GO SUB 8500: PRINT  INK 6;A$
1075 BEEP .05,0
1080 IF K<500 THEN  GO TO 1010
1090 RETURN
1100 REM ***********************
1110 REM   CONFORMA LAS REGLAS
1120 LET R=RU
1130 LET E$=J$( TO R): LET F$=J$
(R+4 TO )
1140 IF E$(1)<>"X" THEN  PRINT
INK 5: FLASH 1;"ERROR EN LA REGL
A": GO TO 70
1150 REM   LA SIGUIENTE LINEA
   DETECTA ENTRADAS COMO :
   X RELACION Z SI X ES Z
1160 IF F$(LEN F$-1 TO )="Z " TH
EN  GO TO 1390
1170 PRINT  INK 7;TAB 4;"COMPILA
CION DE LA REGLA"
1180 FOR T=1 TO 100
1190 LET R$(T)=""
1200 NEXT T
1210 LET E$=E$(3 TO ): LET F$=F$
(3 TO LEN F$)
1220 LET K=0: LET RR=0
1230 LET K=K+1
1240 IF CODE Z$(K)=32 THEN  GO T
O 1300
1245 LET A$=Z$(K): GO SUB 8500
1250 IF A$(LEN A$-LEN F$+1 TO )<
>F$ THEN  GO TO 1370
1260 LET RR=RR+1
1270 LET R$(RR)=A$( TO LEN A$-LE
N F$)+E$+"@"
1280 PRINT "> ";: LET A$=R$(RR):
 GO SUB 8500: PRINT  INK 6;A$
1285 BEEP .1,10: BEEP .3,15
1290 GO TO 1230
1300 IF RR=0 THEN  RETURN
1310 LET RC=0
1320 LET RC=RC+1
1330 LET Z$(K)=R$(RC)
1340 IF K<500 THEN  LET K=K+1
1350 IF RC<RR THEN  GO TO 1320
1360 RETURN
1370 IF K<500 THEN  GO TO 1230
1380 RETURN
```

```
1390 REM REGLA CON 2 VARIABLES
1400 FOR T=1 TO 100
1410 LET R$(T)=""
1420 NEXT T
1430 LET K=0: LET RR=0
1440 IF K=500 THEN  RETURN
1450 LET K=K+1
1460 IF CODE Z$(K)=32 THEN  GO T
O 1770
1470 REM DESCOMPOSICION EN TRES
     PALABRAS
1480 LET Q$=Z$(K)
1490 LET J=0
1500 LET J=J+1
1510 IF Q$(J TO J)=" " THEN  GO
TO 1540
1520 IF J<LEN Q$ THEN  GO TO 150
0
1530 PRINT  INK 5;"ERROR EN COMP
ILACION DE REGLA": GO TO 70
1540 LET A$=Q$( TO J)
1550 LET Q$=Q$(J+1 TO )
1560 LET J=0
1570 LET J=J+1
1580 IF Q$(J)=" " THEN  GO TO 16
10
1590 IF J<LEN Q$ THEN  GO TO 157
0
1600 PRINT  INK 5;"ERROR EN COMP
ILACION DE REGLA": GO TO 70
1610 LET B$=Q$( TO J)
1620 LET Q$=Q$(J+1 TO )
1630 LET J=0
1640 LET J=J+1
1650 IF Q$(J)=" " THEN  GO TO 16
80
1660 IF J<LEN Q$ THEN  GO TO 164
0
1670 PRINT  INK 5;"ERROR EN COMP
ILACION DE REGLA": GO TO 70
1680 PRINT  INK 7;TAB 4;"COMPILA
CION DE LA REGLA"
1690 LET C$=Q$( TO J)
1700 LET M$=F$(3 TO 3+LEN B$-1)
1710 IF B$<>M$ THEN  GO TO 1440
1720 LET RR=RR+1
1730 LET N$=E$(3 TO LEN E$-2)
1740 LET R$(RR)=A$+N$+C$+"@"
1750 PRINT "> ";: LET A$=R$(RR):
 GO SUB 8500: PRINT  INK 6;A$
1755 BEEP .1,10: BEEP .3,15
1760 GO TO 1440
1770 IF RR=0 THEN  RETURN
1780 LET M=0
1790 LET M=M+1
1800 IF M>RR THEN  RETURN
1810 LET Z$(K)=R$(M)
1820 IF K=500 THEN  PRINT  INK 5
: FLASH 1;"MEMORIA LLENA": BEEP
.5,15: GO TO 70
1830 LET K=K+1
1840 GO TO 1790
1850 REM ***********************
1860 REM        LISTA TODO
1870 PRINT
1880 LET K=0
1890 LET K=K+1
1900 IF CODE Z$(K)=32 THEN  RETU
RN
1910 LET A$=Z$(K): GO SUB 8500:
PRINT  INK 6;A$
1915 BEEP .05,0
1920 IF K<500 THEN  GO TO 1890
1930 RETURN
1940 REM ***********************
1950 REM CONFORMA LAS REGLAS CON
     "Y" DEL SIGUIENTE TIPO:
1960 REM (X ENGLOBA Z SI X ES
     UNA COSA Y Z ES OTRA)
1970 REM EL CONCEPTO X DEBE
     ESTAR EN LISTA,PRECEDIENDO
     A Z PARA TODOS LOS EJEMPLOS
     QUE SE CODIFIQUEN
1980 REM DIVIDIR EN SECCIONES
1990 LET J$=J$(2 TO ):
 REM ELIMINA "X"
2000 PRINT  INK 7;TAB 4;"COMPILA
CION DE LA REGLA"
2010 LET J=1
2020 LET J=J+1
2030 IF J$(J)=" " THEN  GO TO 20
60
2040 IF J<LEN J$ THEN  GO TO 202
0
2050 PRINT  INK 5;"ERROR EN COMP
ILACION DE REGLA": RETURN
2060 LET I$=J$( TO J):
 REM RELACION 1
2070 LET J$=J$(J+7 TO ):
 REM CENTRADO EN EL COMIENZO DE
LA SEGUNDA RELACION
2080 LET J=1: LET CONT=0
2090 LET J=J+1
2100 IF J$(J)=" " THEN  LET CONT
=CONT+1
2110 IF CONT=2 THEN  GO TO 2140
2120 IF J<LEN J$ THEN  GO TO 209
0
2130 PRINT  INK 5;"ERROR EN COMP
ILACION DE REGLA": RETURN
2140 LET B$=J$( TO J):
 REM ENUNCIADO 1
2150 LET C$=J$(J+4 TO ):
 REM ENUNCIADO 2
2160 IF C$=" " THEN  PRINT "ERRO
R EN COMPILACION DE REGLA": RETU
```

```
RN
2170 REM AHORA EL PROGRAMA SE
     DIRIGE A LA BASE DE DATOS
2180 FOR T=1 TO 200
2190 LET R$(T)=""
2200 NEXT T
2210 LET R1=0: LET R2=99
2220 LET K=0
2230 LET K=K+1
2240 IF CODE Z$(K)=32 THEN  GO T
O 2310
2250 IF R1=99 OR R2=200 THEN  PR
INT "INSUFICIENCIA DE MEMORIA":
GO TO 2310
2260 LET LB=LEN B$
2270 LET A$=Z$(K): GO SUB 8500:
IF A$(LEN A$-LB+1 TO )=B$ THEN
LET R1=R1+1: LET R$(R1)=A$( TO L
EN A$-LB)+"@"
2280 LET LC=LEN C$
2290 LET A$=Z$(K): GO SUB 8500:
IF A$(LEN A$-LC+1 TO )=C$ THEN
LET R2=R2+1: LET R$(R2)=A$( TO L
EN A$-LC)+"@"
2300 IF K<500 THEN  GO TO 2230
2310 IF CODE R$(100)=32 THEN  PR
INT  INK 5;"EL ENUNCIADO 2 INTRO
DUCIDO NO   CONSTA EN LA BASE DE
 DATOS": BEEP .1,10: BEEP .1,15:
 RETURN
2320 REM CODIFICACION DE REGLAS
2330 LET R1=0: LET R2=99
2340 LET R2=R2+1
2350 LET R1=R1+1
2360 IF R$(R1)=" " OR R$(R2)=" "
 THEN  GO TO 2370
2362 GO SUB 8600: GO SUB 8700
2365 LET Z$(K)=W$+I$+V$+" "+"@"
2370 PRINT "> ";: LET A$=Z$(K):
GO SUB 8500: PRINT  INK 6;A$
2375 BEEP .1,10: BEEP .3,15
2380 LET K=K+1
2390 IF CODE R$(R2+1)<>32 THEN
GO TO 2340
2400 IF CODE R$(R1+1)<>32 THEN
GO TO 2350
2410 RETURN
2420 REM ***********************
2430 REM       ARITMETICA
2440 LET LJ=LEN J$
2450 IF AR<3 THEN  GO SUB 2490
2460 IF AR=3 THEN  GO SUB 2890
2470 IF AR=4 THEN  GO SUB 3080
2480 RETURN
2490 REM ****SUMA - VECES*******
2500 LET J$=J$(6 TO LJ)
2510 IF J$(1)="(" THEN  LET J$=J
$(2 TO )
2520 LET LJ=LEN J$
2530 LET K=0
2540 LET K=K+1
2550 IF J$(K)=" " THEN  LET A$=J
$( TO K-1): LET J$=J$(K+1 TO ):
GO TO 2580
2560 IF K<LJ THEN  GO TO 2540
2570 PRINT  INK 5;TAB 7;"ERROR A
RITMETICO": RETURN
2580 LET LJ=LEN J$
2590 LET K=0
2600 LET K=K+1
2610 IF J$(K)=" " THEN  LET B$=J
$( TO K-1): LET J$=J$(K+1 TO ):
GO TO 2640
2620 IF K<LJ THEN  GO TO 2600
2630 PRINT  INK 5;TAB 7;"ERROR A
RITMETICO": RETURN
2640 LET LJ=LEN J$
2650 LET K=0
2660 LET K=K+1
2670 IF J$(K)=")" THEN  LET C$=J
$( TO K-1): GO TO 2700
2680 IF K<LJ THEN  GO TO 2660
2690 PRINT  INK 5;TAB 2;"ERROR (
DEMASIADAS VARIABLES)": RETURN
2700 LET AN=0: LET BN=0: LET CN=
0
2710 IF CODE A$>57 THEN  LET AN=
1
2720 IF CODE B$>57 THEN  LET BN=
2
2730 IF CODE C$>57 THEN  LET CN=
4
2740 LET GU=AN+BN+CN: IF GU=3 OR
 GU=5 OR GU=6 THEN  GO TO 2690
2750 IF AR=2 THEN  GO TO 2820:
  REM VECES
2760 IF GU>0 THEN  GO TO 2790
2770 IF VAL A$+VAL B$=VAL C$ THE
N  PRINT  INK 6;"SI": BEEP .1,10
: BEEP .3,15: RETURN
2780 PRINT  INK 6;"NO": BEEP .1,
10: BEEP .3,15: RETURN
2790 IF GU=1 THEN  PRINT VAL C$-
VAL B$: GO SUB 40: RETURN
2800 IF GU=2 THEN  PRINT VAL C$-
VAL A$: GO SUB 40: RETURN
2810 PRINT  INK 6;VAL A$+VAL B$:
 GO SUB 40: RETURN
2820 REM *******VECES**********
2830 IF GU>0 THEN  GO TO 2860
2840 IF VAL A$*VAL B$=VAL C$ THE
N  PRINT  INK 6;"SI": BEEP .1,10
: BEEP .3,15: RETURN
2850 PRINT  INK 6;"NO": BEEP .1,
10: BEEP .3,15: RETURN
2860 IF GU=1 THEN  PRINT VAL C$/
```

```
VAL B$: GO SUB 40: RETURN
2870 IF GU=2 THEN  PRINT VAL C$/
VAL A$: GO SUB 40: RETURN
2880 PRINT  INK 6;VAL A$*VAL B$:
 GO SUB 40: RETURN
2890 REM ********MENOR**********
2900 LET NF=0
2910 IF CODE J$<58 THEN  LET NF=
1: REM NUMEROS
2920 LET CONT=0
2930 LET K=0
2940 LET K=K+1
2950 IF J$(K)=" " THEN  LET CONT
=CONT+1
2960 IF CONT=2 THEN  GO TO 3000
2970 IF K<LEN J$ THEN  GO TO 294
0
2980 PRINT  INK 5;TAB 4;"ERROR E
N LA COMPARACION"
2990 RETURN
3000 LET B$=J$(K+1 TO )
3010 LET A$=J$( TO K-6)
3020 IF NF=1 THEN  GO TO 3050
3030 IF A$<B$ THEN  PRINT  INK 6
;"SI": RETURN
3040 PRINT  INK 6;"NO": BEEP .1,
10: BEEP .3,15: RETURN
3050 REM ** NUMEROS **
3060 IF VAL A$<VAL B$ THEN  PRIN
T  INK 6;"SI": BEEP .1,10: BEEP
.3,15: RETURN
3070 PRINT  INK 6;"NO": BEEP .1,
10: BEEP .3,15: RETURN
3080 REM *********ENT***********
3090 IF J$(LEN J$-1 TO )="X " TH
EN  GO TO 3190
3100 LET K=0
3110 LET K=K+1
3120 IF J$(K)=" " THEN  GO TO 31
60
3130 IF K<LEN J$ THEN  GO TO 311
0
3140 PRINT  INK 5;TAB 7;"ERROR A
RITMETICO"
3150 RETURN
3160 LET A=VAL (J$( TO K-1))
3170 IF INT A=A THEN  PRINT  INK
 6;"SI": BEEP .1,10: BEEP .3,15:
 RETURN
3180 PRINT  INK 6;"NO": BEEP .1,
10: BEEP .3,15: RETURN
3190 LET K=0
3200 LET K=K+1
3210 IF J$(K)=" " THEN  GO TO 32
40
3220 IF K<LEN J$ THEN  GO TO 320
0
3230 PRINT  INK 5;TAB 7;"ERROR A
RITMETICO": RETURN
3240 PRINT  INK 6;INT (VAL (J$(
TO K-1)))
3250 RETURN
3260 REM ***********************
3270 REM      INICIALIZACION
3280 CLS
3290 DIM Z$(500,40): DIM R$(200,
40)
3300 RETURN
8500 LET T=1
8505 IF A$(T)="@" THEN  GO TO 85
30
8510 LET D$=A$( TO T)
8520 LET T=T+1: GO TO 8505
8530 LET A$=D$: RETURN
8600 LET B$=R$(R1): LET T=1
8605 IF B$(T)="@" THEN  GO TO 86
30
8610 LET W$=B$( TO T)
8620 LET T=T+1: GO TO 8605
8630 RETURN
8700 LET B$=R$(R2): LET T=1
8705 IF B$(T)="@" THEN  GO TO 87
30
8710 LET V$=B$( TO T)
8720 LET T=T+1: GO TO 8705
8730 RETURN
```

FUZZY RITA

```
  1 REM **************
  2 REM *            *
  3 REM * FUZZY RITA *
  4 REM *            *
  5 REM **************
 10 REM
 20 GO SUB 1380: REM INICIO
 30 GO SUB 1160: REM OPCIONES
DE SALIDA
  40 GO SUB 1250: REM PREGUNTAS
DISCRIMINATORIAS
  50 GO SUB 140: REM CONSULTAS
AL USUARIO
  60 GO SUB 480: REM TOMA DE
DECISION Y ACTUALIZACION DEL
CONOCIMIENTO BASE
```

```
  70 PRINT  INK 6;TAB 3;"PULSE <
ENTER> PARA CONTINUAR"
  80 IF X$<>"" THEN  INPUT  LINE
 I$: GO TO 50
  90 PRINT  INK 6;TAB 8;"ADIESTR
AMIENTO"
 100 PRINT  INK 6;TAB 3;"O CUALQ
UIER TECLA Y DESPUES";TAB 3;"<EN
TER> PARA UTILIZAR RITA"
 110 INPUT  LINE X$: GO TO 50
 120 STOP
 130 REM ********************
 140 REM CONSULTAS AL USUARIO
 150 CLS
 160 PRINT "LOS SIGUIENTES SON L
OS SUJETOS  QUE PUEDO DIFERENCIA
R"
 170 PRINT
 180 FOR J=1 TO TT
 190 PRINT  INK 5;" >  "; INK 7;
A$(J)
 200 NEXT J
 210 GO SUB 1500
 220 IF X$="" THEN  PRINT  INK 6
;"PIENSE EN UNO Y PULSE <ENTER>"
 230 IF X$<>"" THEN  PRINT  INK
6;"ESTOY PREPARADO PARA DETERMIN
AR CUAL HA ELEGIDO"
 240 IF X$="" THEN  INPUT  LINE
J$
 250 LET SUMA=.5
 260 FOR J=1 TO DQ
 270 LET SUMA=SUMA+SUMA
 280 GO SUB 1500
 290 IF X$<>"" AND TT>2 THEN  GO
 TO 390: REM COMPROBACION PARA
 DETERMINAR SI SE PUEDE OMITIR
 LA PREGUNTA
 300 PRINT "INTRODUZCA UN NUMERO
 ENTRE 0 Y 1"
 310 PRINT "0=FALSO,1=VERDAD,$ P
ARA TERMINAR"
 320 PRINT : PRINT  INK 5;E$(J)
 330 INPUT  LINE H$: IF H$="$" T
HEN  PRINT : PRINT "GRACIAS": PR
INT : STOP
 340 LET C(J)=VAL H$
 350 LET C(J)=SUMA*C(J)
 360 NEXT J
 370 RETURN
 380 REM *****************
 390 REM COMPROBACION PARA
DETERMINAR SI LA PREGUNTA SE
PUEDE OMITIR
 400 LET OMITIR=1
 410 FOR W=1 TO TT
 420 IF ABS (B(W,J)-B(1,J))>.7 T
HEN  LET OMITIR=0
 430 NEXT W
 440 IF OMITIR=0 THEN  GO TO 300
 450 LET C(OMITIR)=B(W,J)
 460 GO TO 360
 470 REM ***************
 480 REM TOMA DE DCISION
 490 FOR J=1 TO TT
 500 LET D(J)=0: LET E(J)=0: LET
F(J)=0
 510 NEXT J
 520 LET SUMA=.5
 530 FOR J=1 TO TT
 540 LET SUMA=SUMA+SUMA
 550 FOR X=1 TO DQ
 560 REM PRUEBE CON DISTINTOS
COEFICIENTES EN LAS TRES LINEAS
SIGUIENTES HASTA OBTENER LOS
MEJORES RESULTADOS
 570 IF C(X)=B(J,X) THEN  LET D(
J)=D(J)+1
 580 IF ABS (C(X)-B(J,X))<.6*SUM
A THEN  LET E(J)=E(J)+.4
 590 IF ABS (C(X)-B(J,X))<1.2*SU
MA THEN  LET F(J)=F(J)+.1
 600 NEXT X
 610 NEXT J
 620 LET A1=1: LET A2=1: LET A3=
1
 630 LET F1=1: LET F2=1: LET F3=
1
 640 FOR J=1 TO TT
 650 IF D(J)>F1 THEN  LET F1=D(J
): LET A1=J
 660 IF E(J)>F2 THEN  LET F2=E(J
): LET A2=J
 670 IF F(J)>F3 THEN  LET F3=F(J
): LET A3=J
 680 NEXT J
 690 REM COMUNICACION RESULTADO
 700 PRINT
 710 LET CFLG=0
 720 PRINT "EL RESULTADO MAS PRO
BABLE ES"'TAB 1; INK 5;A$(A1)
 730 IF A2<>A1 THEN  PRINT "EL S
IGUIENTE MAS PROBABLE ES"'TAB 1;
 INK 5;A$(A2): LET CFLG=1
 740 IF A3<>A2 AND A3<>A1 THEN
PRINT "EL SIGUIENTE MAS PROBABLE
 ES"'TAB 1; INK 5;A$(A3): LET CF
LG=2
 750 PRINT
 760 PRINT  INK 6;"ES CORRECTO E
L RESULTADO MAS    PROBABLE? (S/
N)"
 770 INPUT  LINE F$
 780 IF F$<>"S" AND F$<>"N" THEN
  GO TO 770
 790 IF F$="S" AND X$<>"" THEN
```

```
RETURN
 800 IF F$="S" THEN  GO TO 980
 810 IF TT=2 AND A1=1 THEN  LET
A1=2: GO TO 980
 820 IF TT=2 THEN  LET A1=1: GO
TO 980
 830 IF CFLG=0 THEN  GO TO 890
 840 PRINT "ES CORRECTA MI SEGUN
A ELECCION? (S/N)"
 850 INPUT  LINE F$
 860 IF F$="N" THEN  GO TO 890
 870 IF CFLG=1 THEN  LET A1=A2:
GO TO 980
 880 IF CFLG=2 THEN  LET A1=A3:
GO TO 980
 890 GO SUB 1500
 900 FOR J=1 TO TT
 910 PRINT  INK 5;J;"- "; INK 7;
A$(J)
 920 NEXT J
 930 PRINT
 940 PRINT "CUAL ES EL CORRECTO?
"
 950 INPUT A1
 960 IF A1<1 OR A1>TT THEN  GO T
O 950
 970 REM ***********************
 972 REM ADIESTRAMIENTO DE RITA
 976 REM ACTUALIZACION DEL
CONOCIMIENTO BASE
 980 FOR J=1 TO DQ
 990 IF B(A1,J)<>0 THEN  LET B(A
1,J)=(C(J)+5*B(A1,J))/6
1000 IF B(A1,J)=0 THEN  LET B(A1
,J)=C(J)
1010 LET B(A1,J)=INT (.5+(10*B(A
1,J))/10)
1020 NEXT J
1030 PRINT
1040 IF U$="" THEN  RETURN
1050 FOR J=1 TO TT
1060 PRINT : GO SUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K); INK 5;B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
1150 REM ***********************
1160 REM OPCIONES DE SALIDA
1170 LET TT=0
1180 LET TT=TT+1
1190 GO SUB 1500
1200 PRINT  INK 6;"INTRODUZCA LA
 OPCION NUMERO ";TT'" PARA ACABA
R PULSE <ENTER>"
1210 INPUT  LINE A$(TT)
1215 PRINT : PRINT A$(TT)
1220 IF A$(TT,1)=" " OR TT=51 TH
EN  LET TT=TT-1: RETURN
1230 GO TO 1180
1240 REM ***********************
1250 REM
  PREGUNTAS DISCRIMINATORIAS
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 LET DQ=0
1310 LET DQ=DQ+1
1320 GO SUB 1500
1330 PRINT  INK 6;"INTRODUZCA LA
 PREGUNTA NUMERO ";DQ'" PARA ACA
BAR PULSE <ENTER>"
1340 INPUT  LINE E$(DQ)
1345 PRINT : PRINT E$(DQ)
1350 IF E$(DQ,1)=" " OR DQ=51 TH
EN  LET DQ=DQ-1: RETURN
1360 GO TO 1310
1370 REM ***********************
1380 REM INICIALIZACION
1382 POKE 23658,8
1384 POKE 23609,40
1386 POKE 23692,255
1390 BORDER 1: PAPER 1: INK 7: C
LS
1400 REM REDUZCA EN LA LINEA
SIGUIENTE EL TAMANO DE LAS
MATRICES PARA ADAPTARLAS A SUS
NECESIDADES
1410 DIM A$(50,15): DIM B(50,50)
: DIM C(50): DIM D(50): DIM E$(5
0,29): DIM E(50): DIM F(50)
1420 LET X$=""
1430 PRINT "SI QUIERE VER EL CON
OCIMIENTO"
1440 PRINT "BASE ACTUALIZADO,PUL
SE CUALQUIER"
1450 PRINT "TECLA SEGUIDA DE <EN
TER> DESPUESDE CADA EJECUCION"
1460 PRINT : PRINT "SI NO LO DES
EA,PULSE SOLO LA    TECLA <ENTER
>"
1470 INPUT  LINE U$
1480 CLS
1490 RETURN
1500 PRINT  INK 2;"------------
--------------------"
1510 POKE 23692,255
1520 RETURN
```

Programas para Amstrad

SSLISP

```
1 REM *************
2 REM *           *
3 REM * S.S. LISP *
4 REM *           *
5 REM *************
10 REM
20 MODE 1:WINDOW #0,1,40,1,23
25 WINDOW #1,1,40,24,25
26 PAPER #1,1:PEN #1,0
27 CLS #1
30 FIN=1
40 ON ERROR GOTO 3690
50 GOTO 3550:REM INICIALIZACION
60 REM
70 REM    ************************
80 A$=MID$(A$,E):A$=LEFT$(A$,LEN(A$)-1)
90 RETURN
100 REM    ************************
110 PRINT
120 IF FIN=1 THEN FIN=0:PRINT " PARA TER
    MINAR PULSA [ ENTER ]"
130 PRINT STRING$(39,"-")
140 GOSUB 3730:REM INTRODUCCION DE
               MENSAJES
150 NN=0
160 LINE INPUT ": ",A$:A$=UPPER$(A$)
165 CLS #1
170 IF A$="" THEN 3600
180 REM    ************************
190 FOR J=1 TO 12
200 X(J)=0:Y(J)=0:Z(J)=0
210 NEXT J
220 REM    ************************
230 R=0:S=0:T=0:CUNO=0:CDOS=0:ED=0
240 FOR J=1 TO LEN(A$)
250 B$=MID$(A$,J,1)
260 IF B$="(" THEN S=S+1:Z(S)=J:IF T=0
    THEN CUNO=J
270 IF B$=")" THEN T=T+1:Y(T)=J:IF CDOS
    <>0 AND ED=0 THEN ED=J-1
280 IF T=1 AND B$=")" THEN CDOS=J
290 IF B$=" " THEN R=R+1:X(R)=J
300 NEXT J
310 IF S=T THEN 370:REM
    EQUILIBRADO DEL NUMERO DE ( )
320 IF S<T THEN PRINT " ->  OMISION DE
    ("
330 IF S>T THEN PRINT " ->  OMISION DE
    )"
340 INPUT "  +    ",B$
350 A$=A$+B$
360 GOTO 190
370 IF NWDS=0 OR NN=1 THEN 440
380 M$=LEFT$(A$,X(1)-1)
390 FOR J=1 TO NWDS
400 IF M$=N$(J)  THEN  A$=O$(J)+MID$(A$,
    LEN(N$(J))+1)
```

```
410 NEXT J
420 NN=1
430 GOTO 190
440 FLAG=0:B$=" NADA"
450 IF LEFT$(A$,5)="CAR (" THEN FLAG=1
460 IF LEFT$(A$,5)="CDR (" THEN FLAG=2
470 IF LEFT$(A$,6)="CONS (" THEN FLAG=3
480 IF LEFT$(A$,6)="ATOM (" THEN FLAG=4
490 IF LEFT$(A$,7)="IGUAL (" THEN FLAG=5
500 IF LEFT$(A$,6)="NULO (" THEN FLAG=6
510 IF LEFT$(A$,9)="MIEMBRO (" THEN FLAG
    =7
520 IF LEFT$(A$,8)="MIGUAL (" THEN FLAG=
    8
530 IF LEFT$(A$,7)="JUNTA (" THEN FLAG=9

540 IF LEFT$(A$,10)="INVIERTE (" THEN
    FLAG=10
550 IF LEFT$(A$,7)="MISMO (" THEN FLAG=
    11
560 IF LEFT$(A$,7)="LISTA (" THEN FLAG=
    12
570 IF LEFT$(A$,8)="DEFINE (" THEN FLAG=
    13
580 IF LEFT$(A$,6)="INC1 (" THEN FLAG=14
590 IF LEFT$(A$,6)="DEC1 (" THEN FLAG=15
600 IF LEFT$(A$,7)="CERO? (" THEN FLAG=
    16
610 IF LEFT$(A$,7)="RESTA (" THEN FLAG=
    17
620 IF LEFT$(A$,5)="EXP (" THEN FLAG=18
630 IF LEFT$(A$,5)="MAX (" THEN FLAG=19

640 IF LEFT$(A$,5)="MIN (" THEN FLAG=20

650 IF LEFT$(A$,6)="SUMA (" THEN FLAG=21
660 IF LEFT$(A$,7)="MENOS (" THEN FLAG=
    22
670 IF LEFT$(A$,8)="DIVIDE (" THEN FLAG=
    23
680 IF LEFT$(A$,7)="RECIP (" THEN FLAG=
    24
690 IF LEFT$(A$,7)="RESTO (" THEN FLAG=
    25
700 IF LEFT$(A$,6)="MULT (" THEN FLAG=26
710 IF LEFT$(A$,7)="MAYOR (" THEN FLAG=
    27
720 IF LEFT$(A$,7)="MENOR (" THEN FLAG=
    28
730 IF LEFT$(A$,11)="NEGATIVO? (" THEN
    FLAG=29
740 IF LEFT$(A$,9)="NUMERO? (" THEN
    FLAG=30
750 IF LEFT$(A$,6)="UNO? (" THEN FLAG=31
760 IF FLAG>13 THEN 790
770 ON FLAG GOSUB 940,990,1070,1210,1300
    ,1430,1480,1610,1800,1860,2100,2230,
    3400
780 GOTO 830
790 IF FLAG>24 THEN 820
800 ON FLAG-13 GOSUB 2280,2330,2380,2450
    ,2660,2700,2890,2920,2970,3020,3060
810 GOTO 830
820 ON FLAG-24 GOSUB 3130,3170,3220,3260
    ,3300,3350,2380
830 IF FLAG<>0 THEN GOSUB 870:GOTO 110
840 PRINT #1, "  NO ENTIENDO ..."
850 PRINT #1, A$
860 GOTO 110
870 REM ** CONTESTACION DE RETORNO **
880 PRINT "     SU VALOR ES..."
890 IF B$=" C" THEN B$=" C  ( CIERTO )
900 IF B$<>"()" THEN PRINT B$
910 IF B$="()" THEN PRINT "   NADA"
920 RETURN
930 REM *************************
940 REM   **    CAR    **
950 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-
    Z(2))
960 IF S>2 THEN B$=MID$(A$,CUNO,CDOS-
    CUNO+1)
970 RETURN
980 REM *************************
990 REM   **    CDR    **
1000 GOSUB 940
1010 LB=LEN(B$)+7
1020 B$="("+MID$(A$,LB,ED-1)
1030 IF RIGHT$(B$,2)="))" THEN B$=LEFT$
     (B$,LEN(B$)-1)
1040 IF MID$(B$,2,1)=" " THEN B$="("+MID
     $(B$,3)
1050 RETURN
1060 REM *************************
1070 REM   **    CONS    **
1080 B$=MID$(A$,7,LEN(A$)-1)
1090 J=0
1100 IF LEFT$(B$,1)="(" THEN J=1
1110 J=J+1
1120 IF MID$(B$,J,1)="(" THEN 1150
1130 IF J<LEN(B$) THEN 1110
1140 B$="   CONS     INCORRECTO":RETURN
1150 LB=LEN(B$)-1
1160 B$="("+LEFT$(B$,J-1)+MID$(B$,J+1)
1170 B$=LEFT$(B$,LB)
1180 IF RIGHT$(B$,2)=" )" THEN B$=LEFT$
     (B$,LEN(B$)-2)+")"
1190 RETURN
1200 REM *************************
1210 REM   **    ATOM    **
1220 A$=MID$(A$,7,LEN(A$)-1)
1230 J=0
1240 J=J+1
1250 IF MID$(A$,J,1)=" " OR MID$(A$,J,1)
     ="(" THEN RETURN
1260 IF J<LEN(A$) THEN 1240
1270 B$=" C"
1280 RETURN
1290 REM *************************
1300 REM   **    IGUAL    **
1310 E=8:GOSUB 80
1320 J=0
1330 J=J+1
1340 IF MID$(A$,J,1)=")" THEN RETURN
1350 IF MID$(A$,J,1)=" " THEN 1380
1360 IF J<LEN(A$) THEN 1330
1370 RETURN
1380 C$=LEFT$(A$,J-1)
1390 A$=MID$(A$,J+1)
1400 IF C$=A$ THEN B$=" C"
1410 RETURN
1420 REM *************************
1430 REM   **    NULO    **
1440 IF A$="NULO ()" THEN B$="  NULO NE
     CESITA UN ARGUMENTO"
1450 IF A$="NULO (())" THEN B$=" C"
1460 RETURN
1470 REM *************************
1480 REM   **    MIEMBRO    **
1490 C$=MID$(A$,10)
1500 J=1
1510 J=J+1
1520 IF MID$(C$,J,1)=")" OR MID$(C$,J,1)
     ="(" THEN D$=LEFT$(C$,J):GOTO 1550
1530 IF J<LEN(C$) THEN 1510
```

```
1540 RETURN
1550 J=LEN(D$)
1560 J=J+1
1570 IF MID$(C$,J,LEN(D$))=D$  THEN  C$=
     LEFT$(C$,LEN(C$)-1):GOTO 1730
1580 IF J<LEN(C$) THEN 1560
1590 RETURN
1600 REM *************************
1610 REM   **    MIGUAL    **
1620 C$=MID$(A$,9)
1630 J=0
1640 J=J+1
1650 IF MID$(C$,J,1)=" " THEN 1680
1660 IF J<LEN(C$) THEN 1640
1670 RETURN
1680 D$=LEFT$(C$,J)
1690 C$=MID$(C$,J+2)
1700 C$=LEFT$(C$,LEN(C$)-2)+" "
1710 J=0
1720 J=J+1
1730 IF MID$(C$,J,LEN(D$))=D$ THEN B$="(
     "+MID$(C$,J):GOTO 1760
1740 IF J<LEN(C$) THEN 1720
1750 RETURN
1760 B$=LEFT$(B$,LEN(B$)-1)+")"
1770 IF RIGHT$(B$,3)=")))" THEN B$=LEFT$
     (B$,LEN(B$)-1):GOTO 1770
1780 RETURN
1790 REM *************************
1800 REM   **    JUNTA    **
1810 B$=MID$(A$,8)
1820 B$=LEFT$(B$,Y(1)-8)+" "+MID$(B$,Z(3
     )-6)
1830 B$=LEFT$(B$,LEN(B$)-1)
1840 RETURN
1850 REM *************************
1860 REM   **    INVIERTE    **
1870 B$=""
1880 A$=MID$(A$,11):A$=LEFT$(A$,LEN(A$)-
     2)
1890 CT=0
1900 J=0
1910 J=J+1:IF J>LEN(A$) THEN 2020
1920 IF MID$(A$,J,1)=" " THEN 1950
1930 IF MID$(A$,J,1)="(" THEN 1960
1940 GOTO 1910
1950 CT=CT+1:G$(CT)=LEFT$(A$,J-1):A$=MID
     $(A$,J+1):GOTO 1900
1960 J=J+1:IF MID$(A$,J,2)="))"   THEN
     2080
1970 IF MID$(A$,J,1)=")" THEN 2010
1980 IF J=LEN(A$) THEN 2000
1990 GOTO 1960
2000 CT=CT+1:G$(CT)=A$+")":GOTO 2030
2010 CT=CT+1:G$(CT)=LEFT$(A$,J):A$=MID$
     (A$,J+1):GOTO 1900
2020 CT=CT+1:G$(CT)=A$
2030 FOR M=CT TO 1 STEP -1
2040 B$=B$+G$(M):IF M>1 THEN B$=B$+" "
2050 NEXT
2060 B$="("+B$+")"
2070 RETURN
2080 CT=CT+1:G$(CT)=LEFT$(A$,J+1):A$=MID
     $(A$,J+1):GOTO 1900
2090 REM *************************
2100 REM   **    MISMO    **
2110 E=8:GOSUB 80
2120 M=ASC(A$):IF M>47 AND M<58   THEN
     2470
2130 J=0
2140 J=J+1
2150 IF MID$(A$,J,2)=") " THEN J=J+1:
     GOTO 1380
2160 IF MID$(A$,J,3)=")))" THEN 2200
2170 IF MID$(A$,J,2)="))" THEN 2210
2180 IF J<LEN(A$) THEN 2140
2190 RETURN
2200 C$=LEFT$(A$,J+2):A$=MID$(A$,J+4):
     GOTO 1400
2210 C$=LEFT$(A$,J+1):A$=MID$(A$,J+3):
     GOTO 1400
2220 REM *************************
2230 REM   **    LISTA    **
2240 E=8:GOSUB 80
2250 B$="("+A$+")"
2260 RETURN
2270 REM *************************
2280 REM   **    INC1    **
2290 E=7:GOSUB 80
2300 B$=STR$(VAL(A$)+1)
2310 RETURN
2320 REM *************************
2330 REM   **    DEC1    **
2340 E=7:GOSUB 80
2350 B$=STR$(VAL(A$)-1)
2360 RETURN
2370 REM *************************
2380 REM   **    CERO?    **
2390 IF FLAG=16 THEN E=8
2400 IF FLAG=31 THEN E=7
2410 GOSUB 80
2420 IF A$="0" AND FLAG=16 OR A$="1" AND
      FLAG=31 THEN B$=" C"
2430 RETURN
2440 REM *************************
2450 REM   **  DOS ARGUMENTOS  **
2460 E=8:GOSUB 80
2470 J=0
2480 J=J+1
2490 IF MID$(A$,J,1)=" " THEN 2520
2500 IF J<LEN(A$) THEN 2480
2510 B$="* ERROR -    NO SE ADMITE UN
     UNICO        ARGUMENTO *":RETURN
2520 P=VAL(LEFT$(A$,J-1))
2530 Q=VAL(MID$(A$,J+1))
2540 IF FLAG=17 THEN B$=STR$(P-Q):RETURN
2550 IF FLAG=23 OR FLAG=25  THEN B=P/Q
2560 IF FLAG=25 THEN B=INT(0.5+Q*(B-INT(
     B))*1000)/1000
2570 IF FLAG=18 THEN B=P^Q
2580 IF FLAG=11 AND P=Q THEN B$=" C"
2590 IF FLAG=27 AND P>Q THEN B$=" C"
2600 IF FLAG=28 AND P<Q THEN B$=" C"
2610 IF FLAG=32 THEN B=P-Q
2620 IF FLAG=11 OR FLAG>26 THEN RETURN
2630 B$=STR$(B)
2640 RETURN
2650 REM *************************
2660 REM   **    EXP    **
2670 E=6:GOSUB 80
2680 GOTO 2470
2690 REM *************************
2700 REM **  MAX ( MIN SUMA MULT ) **
2710 F$=LEFT$(A$,3):A$=MID$(A$,6)
2720 CT=0:FLAG=0
2730 IF F$="MULT" THEN CT=1
2740 J=0
2750 J=J+1
2760 IF MID$(A$,J,1)=" " THEN 2790
2770 IF J<LEN(A$) THEN 2750
2780 IF J=LEN(A$) THEN FLAG=1
2790 P=VAL(LEFT$(A$,J-1)):IF FLAG=0 THEN
     A$=MID$(A$,J+1)
2800 IF F$<>"SUMA" AND CT=0 THEN CT=P
```

```
2810 IF F$="MAX" AND P>CT THEN CT=P
2820 IF F$="MIN" AND P<CT THEN CT=P
2830 IF F$="SUMA" THEN CT=CT+P
2840 IF F$="MULT" THEN CT=CT*P
2850 IF FLAG=0 THEN 2740
2860 B$=STR$(CT)
2870 RETURN
2880 REM ************************
2890 REM **      MIN      **
2900 GOTO 2710
2910 REM ************************
2920 REM **      SUMA     **
2930 F$="SUMA"
2940 A$=MID$(A$,7)
2950 GOTO 2720
2960 REM ************************
2970 REM **      MENOS    **
2980 E=8:GOSUB 80
2990 B$=STR$(-VAL(A$))
3000 RETURN
3010 REM ************************
3020 REM **      DIVIDE   **
3030 E=9:GOSUB 80
3040 GOTO 2470
3050 REM ************************
3060 REM **      RECIP    **
3070 E=8:GOSUB 80
3080 IF A$="0" THEN B$=" DIVISION POR CE
     RO - NO PERMITIDA":RETURN
3090 B=1/(VAL(A$))
3100 B$=STR$(B)
3110 RETURN
3120 REM ************************
3130 REM **      RESTO    **
3140 E=8:GOSUB 80
3150 GOTO 2470
3160 REM ************************
3170 REM **      MULT     **
3180 F$="MULT"
3190 A$=MID$(A$,7)
3200 GOTO 2720
3210 REM ************************
3220 REM **      MAYOR    **
3230 E=8:GOSUB 80
3240 GOTO 2470
3250 REM ************************
3260 REM **      MENOR    **
3270 E=8:GOSUB 80
3280 GOTO 2470
3290 REM ************************
3300 REM **      NEGATIVO?     **
3310 E=12:GOSUB 80
3320 IF VAL(A$)<0 THEN B$=" C"
3330 RETURN
3340 REM ************************
3350 REM **      NUMERO    **
3360 A$=MID$(A$,10)
3370 IF ASC(A$)>44 AND ASC(A$)<58 THEN
     B$=" C"
3380 RETURN
3390 REM ************************
3400 REM **      DEFINE     **
3410 A$=MID$(A$,9)
3420 F$=LEFT$(A$,X(2)-9)
3430 G$=MID$(A$,X(4)-6)
3440 J=0
3450 J=J+1
3460 IF MID$(G$,J,1)=" " THEN 3490
3470 IF J<LEN(G$) THEN 3450
3480 B$="   ERROR EN LA DEFINICION":
     RETURN
3490 G$=LEFT$(G$,J-1)
3500 NWDS=NWDS+1
3510 O$(NWDS)=G$:N$(NWDS)=F$
3520 B$=F$
3530 RETURN
3540 REM ************************
3550 REM      INICIALIZACION
3560 CLS
3570 DIM G$(20),O$(20),N$(20),X(12),
     Y(12),Z(12)
3580 NWDS=0:REM      CONTADOR DE
          NUEVAS PALABRAS DEFINIDAS
                 POR EL USUARIO
3590 GOTO 110
3600 CLS
3610 LOCATE 3,10
3620 PRINT " QUIERES TERMINAR  [ S , N ]
3630 GOSUB 3770
3640 FIN=1
3650 A$=INKEY$
3660 IF A$="S" THEN CLS:END
3670 IF A$="N" THEN CLS: GOSUB 3810 :
     GOTO 110
3680 GOTO 3650
3690 IF ERR=5 THEN PRINT "  LA ENTRADA
     NO ESTA CORRECTAMENTE ESCRITA":
     GOTO 110
3700 IF ERR=13 THEN PRINT "  CONFUNDES
     LETRAS CON NUMEROS"
3710 PRINT "  LA ENTRADA NO ESTA CORREC
     TAMENTE ESCRITA"
3720 GOTO 110
3730 SOUND 1,100,6,13
3740 SOUND 1,1,10
3750 SOUND 1,100,6,13
3760 RETURN
3770 FOR T=200 TO 100 STEP -5
3780 SOUND 1,T,3
3790 NEXT
3800 RETURN
3810 FOR T=100 TO 200 STEP 5
3820 SOUND 1,T,3
3830 NEXT
3840 RETURN
```

PROLOG-A

```
1 REM ****************
2 REM *              *
3 REM *   PROLOG-A   *
4 REM *              *
5 REM ****************
10 REM
20 REM
30 GOTO 50
40 PRINT " NO HAY ( MAS ) RESPUESTAS .":
   RETURN
```

```
50 GOSUB 3270:REM INICIALIZACION
60 REM ***************************
70 PRINT
80 INPUT "&.",J$:J$=UPPER$(J$)
90 IF J$="" THEN 3310
100 IF J$="LISTA TODO" THEN GOSUB 1860:G
    OTO 70
110 IF LEFT$(J$,6)="LISTA " THEN J$=MID$
    (J$,6)+" ":GOSUB 990:GOTO 70
120 IF RIGHT$(J$,1)<>")" THEN PRINT "1."
    :INPUT M$:J$=J$+M$:GOTO 120
130 LJ=LEN(J$)
140 J$=LEFT$(J$,LJ-1)+" ":REM
    QUITA PARENTESIS FINAL Y LO CAMBIA
    POR UN ESPACIO
150 LJ=LEN(J$)
160 FLAG=0
170 IF LEFT$(J$,5)="METE(" THEN J$=MID$(
    J$,6):FLAG=1
180 RULFLAG=0:MASFLAG=0:ARITFLAG=0
190 FOR R=1 TO LEN(J$)
200 IF MID$(J$,R,4)=" SI " THEN RULFLAG=
    R:FLAG=6
210 IF MID$(J$,R,3)=" Y " THEN MASFLAG=R
220 IF MID$(J$,R,5)="SUMA(" THEN ARITFLA
    G=1
230 IF MID$(J$,R,6)="VECES(" THEN ARITFL
    AG=2
240 IF MID$(J$,R,7)=" MENOR " THEN ARITF
    LAG=3
250 IF MID$(J$,R,3)="ENT" THEN ARITFLAG=
    4
260 NEXT R
270 IF LEFT$(J$,3)="ES(" THEN J$=MID$(J$
    ,4):FLAG=2
280 IF LEFT$(J$,9)="CUAL(X : " THEN J$=M
    ID$(J$,10):FLAG=3
290 IF LEFT$(J$,15)="CUAL((X Z) : X " TH
    EN J$=MID$(J$,16):FLAG=4
300 IF FLAG=0 THEN PRINT"  ERROR DE SINT
    AXIS ":GOTO 70
310 LJ=LEN(J$)
320 REM  AHORA BUSCA LAS SUBRUTINAS
                PERTINENTES
330 IF MASFLAG<>0 THEN GOSUB 1950: GOTO
    70:REM CODIFICA LA REGLA 'Y'
340 IF RULFLAG<>0 AND FLAG<>5 THEN GOSUB
    1110:REM CODIFICA LA REGLA
             CONDICIONAL  'SI'
350 IF ARITFLAG<>0 THEN GOSUB 2430:GOTO
    70:REM OPERACIONES ARITMETICAS
360 IF RIGHT$(J$,3)=" X " OR RIGHT$(J$,3
    )=" Z " THEN J$=LEFT$(J$,LJ-2)+" "
370 LJ=LEN(J$)
380 IF FLAG=1 THEN GOSUB 440:REM METE
390 IF FLAG=2 THEN GOSUB 520:REM  ES
400 IF FLAG=3 THEN GOSUB 610:REM CUAL
410 IF FLAG=4 THEN GOSUB 830:REM CUAL 2

420 GOTO 70
430 REM ***************************
440 REM    METE   ( AFIRMACIONES )
450 K=0
460 K=K+1
470 IF Z$(K)="" THEN Z$(K)=J$:RETURN
480 IF K<1000 THEN 460
490 PRINT " MEMORIA COMPLETA"
500 RETURN
510 REM ***************************
520 REM          ES
530 K=0
540 K=K+1
550 IF Z$(K)="" THEN 580
560 IF Z$(K)=J$ THEN PRINT" SI":GOTO 590
570 IF K<1000 THEN 540
580 PRINT " NO"
590 RETURN
600 REM  ***************************
610 REM  CUAL
    ( BUSCA LAS CORRESPONDENCIAS )
620 IF LEFT$(J$,1)="X" THEN 710
630 J$=LEFT$(J$,LJ-1)
640 K=0
650 K=K+1
660 IF Z$(K)="" THEN 690
670 IF J$=LEFT$(Z$(K),LEN(J$)) THEN PRIN
    T RIGHT$(Z$(K),(LEN(Z$(K))-LEN(J$)))

680 IF K<1000 THEN 650
690 GOSUB 40
700 RETURN
710 REM CUESTIONES QUE COMIENZAN CON X
720 J$=MID$(J$,3,LEN(J$)-3)
730 LJ=LEN(J$)
740 K=0
750 K=K+1
760 IF Z$(K)="" THEN 800
770 Q$=MID$(Z$(K),LEN(Z$(K))-LJ,LJ)
780 IF Q$=J$ THEN PRINT LEFT$(Z$(K),LEN(
    Z$(K))-(LJ+2))
790 IF K<1000 THEN 750
800 GOSUB 40
810 RETURN
820 REM ***************************
830 REM           CUAL 2
840 J$=LEFT$(J$,LJ-2)
850 LJ=LEN(J$)
860 K=0
870 K=K+1
880 IF Z$(K)="" THEN 960
890 LFLAG=0
900 FOR L=1 TO LEN(Z$(K))-LJ
910 IF MID$(Z$(K),L,LJ)=J$ THEN LFLAG=L
920 NEXT L
930 IF LFLAG=0 THEN 950
940 PRINT LEFT$(Z$(K),LFLAG-2);MID$(Z$(K
    ),(LFLAG+LJ))
950 IF K<1000 THEN 870
960 GOSUB 40
970 RETURN
980 REM ***************************
990 REM           LISTADO
1000 K=0
1010 K=K+1
1020 IF Z$(K)="" THEN RETURN
1030 LFLAG=0
1040 FOR L=1 TO LEN(Z$(K))-LEN(J$)
1050 IF MID$(Z$(K),L,LEN(J$))=J$ THEN LF
     LAG=1
1060 NEXT L
1070 IF LFLAG=1 THEN PRINT Z$(K)
1080 IF K<1000 THEN 1010
1090 RETURN
1100 REM ***************************
1110 REM      CONFORMA LAS REGLAS
1120 R=RULFLAG
1130 E$=LEFT$(J$,R):F$=MID$(J$,R+4)
1140 IF LEFT$(E$,1)<>"X" THEN PRINT  "ER
     ROR EN LA REGLA":GOTO 70
1150 REM LA SIGUIENTE LINEA DETECTA
       ENTRADAS  DEL  TIPO:
      X ABSORBE Z SI X ES Z
1160 IF RIGHT$(F$,2)="Z " THEN 1390
1170 PRINT TAB(10);"INTERPRETANDO LA REG
```

```
      LA"
1180 FOR T=1 TO 100
1190 R$(T)=""
1200 NEXT T
1210 E$=MID$(E$,3):F$=MID$(F$,3)
1220 K=0:RR=0
1230 K=K+1
1240 IF Z$(K)="" THEN 1300
1250 IF RIGHT$(Z$(K),LEN(F$))<>F$ THEN 1
      370
1260 RR=RR+1
1270 R$(RR)=LEFT$(Z$(K),(LEN(Z$(K))-LEN(
      F$)))+E$
1280 PRINT "> ";R$(RR)
1290 GOTO 1230
1300 IF RR=0 THEN RETURN
1310 RC=0
1320 RC=RC+1
1330 Z$(K)=R$(RC)
1340 IF K<1000 THEN K=K+1
1350 IF RC<RR THEN 1320
1360 RETURN
1370 IF K<1000 THEN 1230
1380 RETURN
1390 REM * REGLA CON DOS VARIABLES *
1400 FOR T=1 TO 100
1410 R$(T)=""
1420 NEXT T
1430 K=0:RR=0
1440 IF K=1000 THEN RETURN
1450 K=K+1
1460 IF Z$(K)="" THEN 1770
1470 REM  DIVISION EN TRES PALABRAS
1480 Q$=Z$(K)
1490 J=0
1500 J=J+1
1510 IF MID$(Q$,J,1)=" " THEN 1540
1520 IF J<LEN(Q$) THEN 1500
1530 PRINT "ERROR EN LA INTERPRETACION D
      E LA REGLA":GOTO 70
1540 A$=LEFT$(Q$,J)
1550 Q$=MID$(Q$,J+1)
1560 J=0
1570 J=J+1
1580 IF MID$(Q$,J,1)=" " THEN 1610
1590 IF J<LEN(Q$) THEN 1570
1600 PRINT "ERROR EN LA INTERPRETACION D
      E LA REGLA":GOTO 70
1610 B$=LEFT$(Q$,J)
1620 Q$=MID$(Q$,J+1)
1630 J=0
1640 J=J+1
1650 IF MID$(Q$,J,1)=" " THEN 1680
1660 IF J<LEN(Q$) THEN 1640
1670 PRINT "ERROR EN LA INTERPRETACION D
      E LA REGLA":GOTO 70
1680 PRINT TAB(10);"INTERPRETANDO LA REG
      LA"
1690 C$=LEFT$(Q$,J)
1700 M$=MID$(F$,3,LEN(B$))
1710 IF B$<>M$ THEN 1440
1720 RR=RR+1
1730 N$=MID$(E$,3,LEN(E$)-4)
1740 R$(RR)=A$+N$+C$
1750 PRINT "> ";R$(RR)
1760 GOTO 1440
1770 IF RR=0 THEN RETURN
1780 M=0
1790 M=M+1
1800 IF M>RR THEN RETURN
1810 Z$(K)=R$(M)
1820 IF K=1000 THEN PRINT "  NO ESTA EN
```

```
      MEMORIA":GOTO 70
1830 K=K+1
1840 GOTO 1790
1850 REM ***********************
1860 REM       LISTA  TODO
1870 PRINT
1880 K=0
1890 K=K+1
1900 IF Z$(K)="" THEN RETURN
1910 PRINT Z$(K)
1920 IF K<1000 THEN 1890
1930 RETURN
1940 REM ***********************
1950 REM  CONFORMA LAS REGLAS CON 'Y'
        DEL SIGUIENTE TIPO :
1960 REM (X ABSORBE Z SI X (VERBO)
      ALGO 'Y' Z (VERBO) OTRA COSA
1970 REM  EL CONCEPTO X DEBE ESTAR EN
      LISTA, PRECEDIENDO A Z,PARA TODOS
      LOS EJEMPLOS QUE SE
1980 REM  CODIFIQUEN DIVIDIDOS EN
            SECCIONES .
1990 J$=MID$(J$,2):REM  ELIMINA 'X'
2000 PRINT TAB(10);"INTERPRETANDO LA REG
      LA"
2010 J=1
2020 J=J+1
2030 IF MID$(J$,J,1)=" " THEN 2060
2040 IF J< LEN(J$) THEN 2020
2050 PRINT "ERROR EN LA INTERPRETACION D
      E LA REGLA":RETURN
2060 A$=LEFT$(J$,J):REM  RELACCION 1
2070 J$=MID$(J$,J+7):REM  DIVIDE AL
      COMIENZO DE LA SEGUNDA RELACCION
2080 J=1:CNT=0
2090 J=J+1
2100 IF MID$(J$,J,1)=" " THEN CNT=CNT+1
2110 IF CNT=2 THEN 2140
2120 IF J< LEN(J$) THEN 2090
2130 PRINT "ERROR EN LA INTERPRETACION D
      E LA REGLA":RETURN
2140 B$=LEFT$(J$,J):REM  CONCEPTO 1
2150 C$=MID$(J$,J+4):REM  CONCEPTO 2
2160 IF C$=" " THEN PRINT "ERROR EN LA I
      NTERPRETACION DE LA REGLA":RETURN
2170 REM  AHORA VA A LA BASE DE DATOS
2180 FOR T=1 TO 200
2190 R$(T)=""
2200 NEXT T
2210 R1=0:R2=99
2220 K=0
2230 K=K+1
2240 IF Z$(K)="" THEN 2310
2250 IF R1=99 OR R2=200 THEN PRINT " MEM
      ORIA INSUFICIENTE":GOTO 2310
2260 LB=LEN(B$)
2270 IF RIGHT$(Z$(K),LB)=B$ THEN R1=R1+1
      :R$(R1)=LEFT$(Z$(K),LEN(Z$(K))-LB)
2280 LC=LEN(C$)
2290 IF RIGHT$(Z$(K),LC)=C$ THEN R2=R2+1
      :R$(R2)=LEFT$(Z$(K),LEN(Z$(K))-LC)
2300 IF K<1000 THEN 2230
2310 IF R$(100)="" THEN PRINT "EL SEGUND
      O CONCEPTO NO ESTA EN LA BASE  DE D
      ATOS":RETURN
2320 REM  AHORA CODIFICA LAS REGLAS
2330 R1=0:R2=99
2340 R2=R2+1
2350 R1=R1+1
2360 IF R$(R1)>" " AND R$(R2)>" " THEN Z
      $(K)=R$(R1)+A$+R$(R2)+" "
2370 PRINT "> ";Z$(K)
```

```
2380 K=K+1
2390 IF R$(R2+1)<>"" THEN 2340
2400 IF R$(R1+1)<>"' THEN 2350
2410 RETURN
2420 REM ***************************
2430 REM 'OPERACIONES ARITMETICAS
2440 LJ=LEN(J$)
2450 IF ARITFLAG<3 THEN GOSUB 2490
2460 IF ARITFLAG=3 THEN GOSUB 2890
2470 IF ARITFLAG=4 THEN GOSUB 3080
2480 RETURN
2490 REM :***** SUMA - VECES ******
2500 J$=MID$(J$,5,LJ-5)
2510 IF LEFT$(J$,2)="S(" THEN J$=MID$(J$
     ,3) ELSE J$=MID$(J$,2)
2520 LJ=LEN(J$)
2530 K=0
2540 K=K+1
2550 IF MID$(J$,K,1)=" " THEN A$=LEFT$(J
     $,K-1):J$=MID$(J$,K+1):GOTO 2580
2560 IF K<LJ THEN 2540
2570 PRINT TAB(12);" ERROR ARITMETICO":R
     ETURN
2580 LJ=LEN(J$)
2590 K=0
2600 K=K+1
2610 IF MID$(J$,K,1)=" " THEN B$=LEFT$(J
     $,K-1):J$=MID$(J$,K+1):GOTO 2640
2620 IF K<LJ THEN 2600
2630 PRINT TAB(12);" ERROR ARITMETICO":R
     ETURN
2640 LJ=LEN(J$)
2650 K=0
2660 K=K+1
2670 IF MID$(J$,K,1)=")" THEN C$=LEFT$(J
     $,K-1):GOTO 2700
2680 IF K<LJ THEN 2660
2690 PRINT TAB(5);" ERROR (DEMASIADAS VA
     RIABLES)":RETURN
2700 AN=0:BN=0:CN=0
2710 IF ASC(A$)>58 THEN AN=1
2720 IF ASC(B$)>58 THEN BN=2
2730 IF ASC(C$)>58 THEN CN=4
2740 GUIA=AN+BN+CN:IF GUIA=3 OR GUIA=5
     OR GUIA=6 THEN 2690
2750 IF ARITFLAG=2 THEN 2820:REM VECES
2760 IF GUIA>0 THEN 2790
2770 IF VAL(A$)+VAL(B$)=VAL(C$) THEN PRI
     NT " SI":RETURN
2780 PRINT " NO":RETURN
2790 IF GUIA=1 THEN PRINT VAL(C$)-VAL(B$
     ):GOSUB 40:RETURN
2800 IF GUIA=2 THEN PRINT VAL(C$)-VAL(A$
     ):GOSUB 40:RETURN
2810 PRINT VAL(A$)+VAL(B$):GOSUB 40:RETU
     RN
2820 REM **    VECES    **
2830 IF GUIA>0 THEN 2860
2840 IF VAL(A$)*VAL(B$)=VAL(C$) THEN PRI
     NT " SI":RETURN
2850 PRINT " NO":RETURN
2860 IF GUIA=1 THEN PRINT VAL(C$)/VAL(B$
     ):GOSUB 40:RETURN
2870 IF GUIA=2 THEN PRINT VAL(C$)/VAL(A$
     ):GOSUB 40:RETURN
2880 PRINT VAL(A$)*VAL(B$):GOSUB 40:RETU
     RN
2890 REM *******    MENOR    *******
2900 NF=0
2910 IF ASC(J$)<58 THEN NF=1:  REM
     SON NUMEROS
2920 CNT=0
2930 K=0
2940 K=K+1
2950 IF MID$(J$,K,1)=" " THEN CNT=CNT+1
2960 IF CNT=2 THEN 3000
2970 IF K<LEN(J$) THEN 2940
2980 PRINT TAB(10);"ERROR EN LA COMPARAC
     ION"
2990 RETURN
3000 B$=MID$(J$,K+1)
3010 A$=LEFT$(J$,K-6)
3020 IF NF=1 THEN 3050
3030 IF A$<B$ THEN PRINT " SI":RETURN
3040 PRINT " NO":RETURN
3050 REM *    NUMEROS    *
3060 IF VAL(A$)<VAL(B$) THEN PRINT " SI"
     :RETURN
3070 PRINT " NO":RETURN
3080 REM **  ENT ( PARTE ENTERA )  **
3090 IF RIGHT$(J$,2)="X " THEN 3190
3100 K=0
3110 K=K+1
3120 IF MID$(J$,K,1)=" " THEN 3160
3130 IF K<LEN(J$) THEN 3110
3140 PRINT TAB(20);"ERROR ARITMETICO"
3150 RETURN
3160 A=VAL(LEFT$(J$,K-1))
3170 IF INT(A)=A THEN PRINT " SI":RETURN
3180 PRINT " NO":RETURN
3190 K=0
3200 K=K+1
3210 IF MID$(J$,K,1)=" " THEN 3240
3220 IF K<LEN(J$) THEN 3200
3230 PRINT  TAB(20) ;"ERROR ARITMETICO":
     RETURN
3240 PRINT INT(VAL(LEFT$(J$,K-1)))
3250 RETURN
3260 REM *****************************
3270 REM         INICIALIZACION
3280 CLS
3290 DIM Z$(1000),R$(200)
3300 RETURN
3310 CLS
3320 LOCATE 3,10:PRINT "QUIERES TERMINAR
     ?  [S,N]"
3330 A$=INKEY$
3340 IF A$="S" THEN CLS:END
3350 IF A$="N" THEN CLS:GOTO 70
3360 GOTO 3330
```

FUZZY RITA

```
1 REM **************
2 REM *            *
3 REM * FUZZY RITA *
4 REM *            *
5 REM **************
10 REM
20 GOSUB 1420:REM INICIALIZACION
30 GOSUB 1190:REM OPCIONES DE SALIDA
40 GOSUB 1280:REM    PREGUNTAS
        DISCRIMINATORIAS
```

```
50 GOSUB 140:REM PREGUNTAS AL USUARIO
60 GOSUB 500: REM TOMA DE DECISION Y
     ACTUALIZAION DE CONOCIMIENTOS
70 PRINT"PULSA [ ENTER ] PARA CONTINUAR"
80 IF X$<>"" THEN INPUT I$:GOSUB 1580:GO
TO 50
90 PRINT" EL ADIESTRAMIENTO ,"
100 PRINT"O CUALQUIER TECLA SEGUIDA DE
    [ ENTER ]   PARA UTILIZAR A RITA."
110 INPUT X$:GOSUB 1580:GOTO 50
120 END
130 REM *************************
140 REM PREGUNTAS AL USUARIO
150 WINDOW #1,1,40,1,TT+3
160 WINDOW #0,1,40,TT+4,25
170 CLS #0:CLS #1
180 PRINT #1,"ESTOS SON LOS TEMAS ENTRE
    LOS QUE PUEDO DISTIGUIR :"
190 PRINT #1
200 FOR J=1 TO TT
210 PRINT #1," -  ";A$(J)
220 NEXT J
230 GOSUB 1560
240 IF X$="" THEN PRINT"PIENSA UNO Y DES
    PUES PULSA [ ENTER ]."
250 IF X$<>"" THEN PRINT" ESTOY LISTA PA
    RA DEDUCIR CUAL TIENES EN MENTE."
260 IF X$="" THEN INPUT J$
270 ADD=0.5
280 FOR J=1 TO DQ
290 ADD=ADD+ADD
300 GOSUB 1560
310 IF X$<>"" AND TT>2 THEN 410:REM
      REVISION POR SI LA PREGUNTA
         SE PUEDE SALTAR
320 PRINT" PULSA UNO DE ESTOS NUMEROS :"
330 PRINT" 1 (CIERTO)   0 (FALSO)   $ [
    TERMINAR ]"
340 PRINT:PRINT E$(J);
350 INPUT H$:GOSUB 1610:IF H$="$" THEN
    PRINT:PRINT"   HASTA PRONTO":PRINT:
    END
360 C(J)=VAL(H$)
370 C(J)=ADD*C(J)
380 NEXT J
390 RETURN
400 REM *************************
410 REM DECIDE SI LA PREGUNTA
        SE PUEDE SALTAR
420 SALTO=1
430 FOR W=1 TO TT
440 IF ABS(B(W,J)-B(1,J))>0.7 THEN SALTO
    =0
450 NEXT W
460 IF SALTO=0 THEN 320
470 C(J)=B(W,J)
480 GOTO 380
490 REM *************************
500 REM TOMA DE DECISION
510 FOR J=1 TO TT
520 D(J)=0:E(J)=0:F(J)=0
530 NEXT J
540 ADD=0.5
550 FOR J=1 TO TT
560 ADD=ADD+ADD
570 FOR X=1 TO DQ
580 REM JUEGA CON LOS VALORES DE LAS
    TRES LINEAS SIGUIENTES PARA
     LOGRAR UNA MAYOR EFICACIA
590 IF C(X)=B(J,X) THEN D(J)=D(J)+1
600 IF ABS(C(X)-B(J,X))<0.6*ADD THEN E(J
    )=E(J)+0.4
610 IF ABS(C(X)-B(J,X))<1.2*ADD THEN F(J
    )=F(J)+0.3
620 NEXT X
630 NEXT J
640 A1=1:A2=1:A3=1
650 F1=1:F2=1:F3=1
660 FOR J=1 TO TT
670 IF D(J)>F1 THEN F1=D(J):A1=J
680 IF D(J)>F2 THEN F2=E(J):A2=J
690 IF D(J)>F3 THEN F3=F(J):A3=J
700 NEXT J
710 REM ** PRESENTA EL RESULTADO **
720 PRINT
730 CFLG=0
740 PRINT"EL RESULTADO MAS PROBABLE ES "
    ;A$(A1)
750 IF A2<>A1 THEN PRINT"EL SIGUIENTE
    MAS PROBABLE ES ";A$(A2):CFLG=1
760 IF A3<>A2 THEN PRINT"EL SIGUIENTE
    MAS PROBABLE ES ";A$(A3):CFLG=2
770 PRINT
780 PRINT"ES CORRECTO EL RESULTADO MAS
    PROBABLE ?";SPC(10);" S o N ";
790 INPUT F$
800 IF F$<>"S" AND F$<>"N" THEN 790
810 IF F$="S" THEN PER=50 ELSE PER=480
820 IF F$="S" AND X$<>"" THEN RETURN
830 IF F$="S" THEN 1010
840 IF TT=2 AND A1=1 THEN A1=2:GOTO 1010
850 IF TT=2 THEN A1=1:GOTO 1010
860 IF CFLG=0 THEN 920
870 PRINT" ES CORRECTA MI SEGUNDA ELECCI
    ON ? ";SPC(15);" S o N ";
880 INPUT F$
890 IF F$="N" THEN 920
900 IF CFLG=1 THEN A1=A2:GOTO 1010
910 IF CFLG=2 THEN A1=A3:GOTO 1010
920 GOSUB 1560
930 FOR J=1 TO TT
940 PRINT J;"- ";A$(J)
950 NEXT J
960 PRINT
970 PRINT" CUAL ES LA SOLUCION CORRECTA
    ";
980 INPUT A1:GOSUB 1610
990 IF A1<1 OR A1>TT THEN 980
1000 REM ** EDUCANDO A RITA **
     ( ACTUALIZACION DE CONOCIMIENTOS )
1010 FOR J=1 TO DQ
1020 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+
     5.5*B(A1,J))/6
1030 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1040 B(A1,J)=INT(10*B(A1,J))/10
1050 NEXT J
1060 PRINT
1070 IF U$="" THEN RETURN
1080 FOR J=1 TO TT
1090 PRINT:GOSUB 1560
1100 PRINT A$(J)
1110 PRINT
1120 FOR K=1 TO DQ
1130 PRINT E$(K);B(J,K)
1140 NEXT K
1150 NEXT J
1160 PRINT
1170 RETURN
1180 REM *************************
1190 REM  OPCIONES DE SALIDA
1200 TT=0
1210 TT=TT+1
1220 GOSUB 1560
1230 PRINT"INTRODUCE LA RESPUESTA NUMERO
```

```
     ";TT;"    PULSA [ ENTER ] PARA TERMI
     NAR"
1240 INPUT A$(TT):PRINT CHR$(7)
1250 IF A$(TT)="" OR TT=50 THEN TT=TT-1:
     RETURN
1260 GOTO 1210
1270 REM **************************
1280 REM PREGUNTAS DISCRIMINATORIAS
1290 WINDOW #1,1,40,1,TT:CLS #1
1300 WINDOW #0,1,40,TT+1,25:CLS #0
1310 FOR J=1 TO TT
1320 PRINT #1, A$(J)
1330 NEXT J
1340 DQ=0
1350 DQ=DQ+1
1360 GOSUB 1560
1370 PRINT"       INTRODUCE LA PREGUNTA NU
     MERO ";DQ;"   PULSA [ ENTER ] SI NO
      HAY MAS."
1380 INPUT E$(DQ):PRINT CHR$(7)
1390 IF E$(DQ)="" OR DQ=50 THEN DQ=DQ-1:
     RETURN
1400 GOTO 1350
1410 REM **************************
1420 REM  INICIALIZACION
1430 CLS
1440 REM REDUCE LAS MATRICES DE LA
     SIGUIENTE LINEA DE ACUERDO A
          TUS NECESIDADES
1450 DIM A$(50),B(50,50),C(50),D(50),
     E(50),E$(50),F(50)
1460 X$="":MODE 1:BORDER 11
1470 PEN #0,3:PEN #1,0:INK 3,21:INK 0,3
1480 PAPER #0,0:PAPER #1,3:CLS #1:CLS
1490 LOCATE 1,5:PRINT" PULSA UNA TECLA Y
     DESPUES [ ENTER ] ,"
1500 PRINT:PRINT" SI QUIERES VER LOS CON
     OCIMIENTOS"
1510 PRINT:PRINT" ADQUIRIDOS, DESPUES DE
     CADA RONDA ; "
1520 PRINT:PRINT"EN CASO CONTRARIO PULSA
     SOLO [ ENTER ] ."
1530 PRINT:INPUT U$:GOSUB 1580
1540 CLS
1550 RETURN
1560 PRINT STRING$(39,"-")
1570 RETURN
1580 ENV 2,1,49,1,49,-1,1
1590 SOUND 2,60,50,0,2
1600 RETURN
1610 SOUND 2,100,50,0
1620 RETURN
1630 SOUND 1,478,15,10
1640 SOUND 2,239,15,10
1650 SOUND 4,119,15,10
1660 RETURN r,w,,
```

Programas para Commodore 64

La versión del programa SSLISP para Commodore 64 utiliza las siguientes órdenes:

MSX	*COMMODORE*	*MSX*	*COMMODORE*	*MSX*	*COMMODORE*
CAR	CAR	LISTA	LIST	DIVIDE	QUOTIENT
CDR	CDR	DEFINE	DEFINE	RECIP	RECIP
CONS	CONS	INC1	ADD1	RESTO	REMAINDER
ATOM	ATOM	DEC1	SUB1	MULT	TIMES
IGUAL	EQ	CERO?	ZEROP	MAYOR	GREATERP
NULO	NULL	RESTA	DIFFERENCE	MENOR	LESSP
MIEMBRO	MEMBER	EXP	EXPT	NEGATIVO?	MINUSP
MIGUAL	MEMQ	MAX	MAX	NUMERO?	NUMBERP
JUNTA	APPEND	MIN	MIN	UNO?	ONEP
INVIERTE	REVERSE	SUMA	PLUS		
MISMO	EQUAL	MENOS	MINUS		

SSLISP

```
1 REM *****************
2 REM *               *
3 REM *   S.S. LISP   *
4 REM *               *
5 REM *****************
10 REM
20 GOTO 3450
30 REM *************************
40 A$=MID$(A$,E):A$=LEFT$(A$,LEN(A$)-1)
50 RETURN
60 REM *************************
70 PRINT ""
80 NN=0
90 A$="":INPUT":        ";A$
100 IF A$="" THEN END:REM PARA FINALIZAR
 BASTA CON PULSAR <RETURN>
110 REM ************************
120 PRINT "";:FOR J=1 TO 12
130 X(J)=0:Y(J)=0:Z(J)=0
140 NEXT J
150 REM ************************
160 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
170 FOR J=1 TO LEN(A$)
180 B$=MID$(A$,J,1)
190 IF B$="(" THEN S=S+1:Z(S)=J:IF T=0 T
HEN CFIRST=J
200 IF B$=")" THEN T=T+1:Y(T)=J:IF CSECN
D<>0 AND EDGE=0 THEN EDGE=J-1
210 IF T=1 AND B$=")" THEN CSECND=J
220 IF B$=" " THEN R=R+1:X(R)=J
230 NEXT J
240 IF S=T THEN 300:REM () EQUILIBRADO
250 IF S<T THEN PRINT:PRINT " -> OMISION
 DE (":PRINT
260 IF S>T THEN PRINT:PRINT " -> OMISION
 DE )":PRINT
270 INPUT "+ ";B$
280 A$=A$+B$
290 GOTO 120
300 IF NWDS=0 OR NN=1 THEN 370
310 M$=LEFT$(A$,X(1)-1)
320 FOR J=1 TO NWDS
330 IF M$=N$(J) THEN A$=O$(J)+MID$(A$,LE
N(N$(J))+1)
340 NEXT J
350 NN=1
360 GOTO 120
370 FLAG=0:B$="FALSO"
380 IF LEFT$(A$,5)="CAR (" THEN FLAG=1
390 IF LEFT$(A$,5)="CDR (" THEN FLAG=2
400 IF LEFT$(A$,6)="CONS (" THEN FLAG=3
410 IF LEFT$(A$,6)="ATOM (" THEN FLAG=4
420 IF LEFT$(A$,4)="EQ (" THEN FLAG=5
430 IF LEFT$(A$,6)="NULL (" THEN FLAG=6
440 IF LEFT$(A$,8)="MEMBER (" THEN FLAG=
7
450 IF LEFT$(A$,6)="MEMQ (" THEN FLAG=8
460 IF LEFT$(A$,8)="APPEND (" THEN FLAG=
9
470 IF LEFT$(A$,9)="REVERSE (" THEN FLAG
=10
480 IF LEFT$(A$,7)="EQUAL (" THEN FLAG=1
1
490 IF LEFT$(A$,6)="LIST (" THEN FLAG=12
500 IF LEFT$(A$,8)="DEFINE (" THEN FLAG=
13
510 IF LEFT$(A$,6)="ADD1 (" THEN FLAG=14
520 IF LEFT$(A$,6)="SUB1 (" THEN FLAG=15
530 IF LEFT$(A$,7)="ZEROP (" THEN FLAG=1
6
540 IF LEFT$(A$,12)="DIFFERENCE (" THEN
FLAG=17
550 IF LEFT$(A$,6)="EXPT (" THEN FLAG=18
560 IF LEFT$(A$,5)="MAX (" THEN FLAG=19
570 IF LEFT$(A$,5)="MIN (" THEN FLAG=20
580 IF LEFT$(A$,6)="PLUS (" THEN FLAG=21
590 IF LEFT$(A$,7)="MINUS (" THEN FLAG=2
2
600 IF LEFT$(A$,10)="QUOTIENT (" THEN FL
AG=23
610 IF LEFT$(A$,7)="RECIP (" THEN FLAG=2
4
620 IF LEFT$(A$,11)="REMAINDER (" THEN F
LAG=25
630 IF LEFT$(A$,7)="TIMES (" THEN FLAG=2
6
640 IF LEFT$(A$,10)="GREATERP (" THEN FL
AG=27
650 IF LEFT$(A$,7)="LESSP (" THEN FLAG=2
8
660 IF LEFT$(A$,8)="MINUSP (" THEN FLAG=
29
670 IF LEFT$(A$,9)="NUMBERP (" THEN FLAG
=30
680 IF LEFT$(A$,6)="ONEP (" THEN FLAG=31
690 IF FLAG>13 THEN 720
700 ON FLAG GOSUB 840,890,970,1110,1200,
1330,1380,1510,1700,1760,2000,2130,3300
710 GOTO 760
720 IF FLAG>24 THEN 750
730 ON (FLAG-13) GOSUB 2180,2230,2280,23
50,2560,2600,2790,2820,2870,2920,2960
740 GOTO 760
750 ON (FLAG-24) GOSUB 3030,3070,3120,31
60,3200,3250,2280
760 IF FLAG<>0 THEN GOSUB 780
770 GOTO 70
780 REM ***CONTESTACION***
790 GOSUB 4000:PRINT:PRINT"    SU VALOR E
S...":PRINT
800 IF B$<>"()" THEN PRINT"    ";B$
810 IF B$="()" THEN PRINT"    FALSO"
820 RETURN
830 REM ***********************
840 REM ***CAR***
850 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z
(2))
860 IF S>2 THEN B$=MID$(A$,CFIRST,CSECND
-CFIRST+1)
870 RETURN
880 REM ***********************
890 REM ***CDR***
900 GOSUB 840
910 LB=LEN(B$)+7
920 B$="("+MID$(A$,LB,EDGE-1)
930 IF RIGHT$(B$,2)="))" THEN B$=LEFT$(B
$,LEN(B$)-1)
940 IF MID$(B$,2,1)=" " THEN B$="("+MID$
(B$,3)
950 RETURN
960 REM ***********************
970 REM ***CONS***
980 B$=MID$(A$,7,LEN(A$)-1)
990 J=0
1000 IF LEFT$(B$,1)="(" THEN J=1
```

```
1010 J=J+1
1020 IF MID$(B$,J,1)="(" THEN 1050
1030 IF K<LEN(B$) THEN 1010
1040 B$="     > ERROR EN CONS  <":RETURN
1050 LB=LEN(B$)-1
1060 B$="("+LEFT$(B$,J-1)+MID$(B$,J+1)
1070 B$=LEFT$(B$,LB)
1080 IF RIGHT$(B$,2)=" )" THEN B$=LEFT$(
B$,LEN(B$)-2)+")"
1090 RETURN
1100 REM **********************
1110 REM ***ATOM***
1120 A$=MID$(A$,7,LEN(A$)-1)
1130 J=0:B$="FALSO"
1140 J=J+1
1150 IF MID$(A$,J,1)=" " OR MID$(A$,J,1)
="(" THEN RETURN
1160 IF J<LEN(A$) THEN 1140
1170 B$="V"
1180 RETURN
1190 REM **********************
1200 REM ***EQ***
1210 E=5:GOSUB 40
1220 J=0
1230 J=J+1
1240 IF MID$(A$,J,1)=")" THEN RETURN
1250 IF MID$(A$,J,1)=" " THEN 1280
1260 IF J<LEN(A$) THEN 1230
1270 RETURN
1280 C$=LEFT$(A$,J-1)
1290 A$=MID$(A$,J+1)
1300 IF C$=A$ THEN B$="V"
1310 RETURN
1320 REM **********************
1330 REM ***NULL***
1340 IF A$="NULL ()" THEN B$=" *UTILIZAC
ION ILEGAL:NULL REQUIERE ARGUMENTO *"
1350 XX$="NULL (())":IF A$=XX$ THEN B$="
V"
1360 RETURN
1370 REM **********************
1380 REM ***MEMBER***
1390 C$=MID$(A$,9)
1400 J=1
1410 J=J+1
1420 IF MID$(C$,J,1)=")" OR MID$(C$,J,1)
="(" THEN D$=LEFT$(C$,J):GOTO1450
1430 IF J<LEN(C$) THEN 1410
1440 RETURN
1450 J=LEN(D$)
1460 J=J+1
1470 IF MID$(C$,J,LEN(D$))=D$ THEN C$=LE
FT$(C$,LEN(C$)-1):GOTO1630
1480 IF J<LEN(C$) THEN 1460
1490 RETURN
1500 REM ***********************
1510 REM ***MEMQ***
1520 C$=MID$(A$,7)
1530 J=0
1540 J=J+1
1550 XX$=" ":IF MID$(C$,J,1)=XX$ THEN 15
80
1560 IF J<LEN(A$) THEN 1540
1570 RETURN
1580 D$=LEFT$(C$,J)
1590 C$=MID$(C$,J+2)
1600 C$=LEFT$(C$,LEN(C$)-2)+" "
1610 J=0
1620 J=J+1
1630 IF MID$(C$,J,LEN(D$))=D$ THEN B$="(
"+MID$(C$,J):GOTO 1660
1640 IF J<LEN(C$) THEN 1620
1650 RETURN
1660 B$=LEFT$(B$,LEN(B$)-1)+")"
1670 IF RIGHT$(B$,3)=")))" THEN B$=LEFT$
(B$,LEN(B$)-1):GOTO 1670
1680 RETURN
1690 REM ***********************
1700 REM ***APPEND***
1710 B$=MID$(A$,9)
1720 B$=LEFT$(B$,Y(1)-9)+" "+MID$(B$,Z(3
)-7)
1730 B$=LEFT$(B$,LEN(B$)-1)
1740 RETURN
1750 REM ***********************
1760 REM ***REVERSE***
1770 B$=""
1780 A$=MID$(A$,11):A$=LEFT$(A$,LEN(A$)-
2)
1790 CT=1
1800 J=0
1810 J=J+1:IF J>LEN(A$) THEN 1920
1820 XX$=MID$(A$,J,1):IF XX$=" " THEN GO
TO1850
1830 IF XX$="(" THEN 1860
1840 GOTO 1810
1850 CT=CT+1:G$(CT)=LEFT$(A$,J-1):A$=MID
$(A$,J+1):GOTO1800
1860 J=J+1:IF MID$(A$,J,2)="))" THEN 198
0
1870 IF MID$(A$,J,1)=")" THEN 1910
1880 IF J=LEN(A$) THEN 1900
1890 GOTO 1860
1900 CT=CT+1:G$(CT)=A$+")":GOTO 1930
1910 CT=CT+1:G$(CT)=LEFT$(A$,J):A$=MID$(
A$,J+1):GOTO1800
1920 CT=CT+1:G$(CT)=A$
1930 FOR M=CT TO 1 STEP -1
1940 B$=B$+G$(M):IF M>1 THEN B$=B$+" "
1950 NEXT M
1960 B$="("+B$+")"
1970 RETURN
1980 CT=CT+1:G$(CT)=LEFT$(A$,J+1):A$=MID
$(A$,J+2):GOTO 1800
1990 REM ************************
2000 REM ***EQUAL***
2010 E=8:GOSUB 40
2020 M=ASC(A$):IF M>47 AND M<58 THEN 237
0
2030 J=0
2040 J=J+1
2050 IF MID$(A$,J,2)=") " THEN J=J+1:GOT
O 1280
2060 IF MID$(A$,J,3)=")))" THEN 2100
2070 IF MID$(A$,J,2)="))" THEN 2110
2080 IF J<LEN(A$) THEN 2040
2090 RETURN
2100 C$=LEFT$(A$,J+2):A$=MID$(A$,J+4):GO
TO1300
2110 C$=LEFT$(A$,J+1):A$=MID$(A$,J+3):GO
TO1300
2120 REM ************************
2130 REM ***LIST***
2140 E=7:GOSUB 40
2150 B$="("+A$+")"
2160 RETURN
2170 REM ************************
2180 REM ***ADD1***
2190 E=7:GOSUB 40
2200 B$=STR$(VAL(A$)+1)
2210 RETURN
2220 REM ************************
2230 REM ***SUB1***
2240 E=7:GOSUB 40
```

```
2250 B$=STR$(VAL(A$)-1)
2260 RETURN
2270 REM ***********************
2280 REM ***ZEROP***
2290 IF FLAG=16 THEN E=8
2300 IF FLAG=31 THEN E=7
2310 GOSUB 40
2320 IF (A$="0" AND FLAG=16) OR (A$="1"
AND FLAG=31) THEN B$="V"
2330 RETURN
2340 REM ***********************
2350 REM ***DOS ARGUMENTOS***
2360 E=13:GOSUB 40
2370 J=0
2380 J=J+1
2390 IF MID$(A$,J,1)=" " THEN 2420
2400 IF J<LEN(A$) THEN 2380
2410 B$=" * ERROR : NO SE ADMITE UN UNIC
O AR- GUMENTO *":RETURN
2420 P=VAL(LEFT$(A$,J-1))
2430 Q=VAL(MID$(A$,J+1))
2440 IF FLAG=17 THEN B$=STR$(P-Q):RETURN
2450 IF FLAG=23 OR FLAG=25 THEN B=P/Q
2460 IF FLAG=25 THEN B=INT(.5+Q*(B-INT(B
))*1000)/1000
2470 IF FLAG=18 THEN B=P^Q
2480 IF FLAG=11 AND P=Q THEN B$="V"
2490 IF FLAG=27 AND P>Q THEN B$="V"
2500 IF FLAG=28 AND P<Q THEN B$="V"
2510 IF FLAG=32 THEN B=P-Q
2520 IF FLAG=11 OR FLAG>26 THEN RETURN
2530 B$=STR$(B)
2540 RETURN
2550 REM ***********************
2560 REM ***EXPT***
2570 E=7:GOSUB 40
2580 GOTO 2370
2590 REM ***********************
2600 REM ***MAX (MIN PLUS TIMES) ***
2610 F$=LEFT$(A$,3):A$=MID$(A$,6)
2620 CT=0:FLAG=0
2630 IF F$="TIMES" THEN CT=1
2640 J=0
2650 J=J+1
2660 IF MID$(A$,J,1)=" " THEN 2690
2670 IF J<LEN(A$) THEN 2650
2680 IF J=LEN(A$) THEN FLAG=1
2690 P=VAL(LEFT$(A$,J-1)):IF FLAG=0 THEN
 A$=MID$(A$,J+1)
2700 IF F$<>"PLUS" AND CT=0 THEN CT=P
2710 IF F$="MAX" AND P>CT THEN CT=P
2720 IF F$="MIN" AND P<CT THEN CT=P
2730 IF F$="PLUS" THEN CT=CT+P
2740 IF F$="TIMES" THEN CT=CT*P
2750 IF FLAG=0 THEN 2640
2760 B$=STR$(CT)
2770 RETURN
2780 REM ***********************
2790 REM ***MIN***
2800 GOTO 2610
2810 REM ***********************
2820 REM ***PLUS***
2830 F$="PLUS"
2840 A$=MID$(A$,7)
2850 GOTO 2620
2860 REM ***********************
2870 REM ***MINUS***
2880 E=8:GOSUB 40
2890 B$=STR$(-VAL(A$))
2900 RETURN
2910 REM ***********************
2920 REM ***QUOTIENT***
2930 E=11:GOSUB 40
2940 GOTO 2370
2950 REM ***********************
2960 REM ***RECIP***
2970 E=8:GOSUB 40
2980 IF B$="0" THEN B$=" ERROR : DIVISIO
N POR CERO":RETURN
2990 B=1/(VAL(A$))
3000 B$=STR$(B)
3010 RETURN
3020 REM ***********************
3030 REM ***REMAINDER***
3040 E=12:GOSUB 40
3050 GOTO 2370
3060 REM ***********************
3070 REM ***TIMES***
3080 F$="TIMES"
3090 A$=MID$(A$,8)
3100 GOTO 2620
3110 REM ***********************
3120 REM ***GREATERP***
3130 E=11:GOSUB 40
3140 GOTO 2370
3150 REM ***********************
3160 REM ***LESSP***
3170 E=8:GOSUB 40
3180 GOTO 2370
3190 REM ***********************
3200 REM ***MINUSP***
3210 E=9:GOSUB 40
3220 IF VAL(A$)<0 THEN B$="V"
3230 RETURN
3240 REM ***********************
3250 REM ***NUMBERP***
3260 A$=MID$(A$,10)
3270 IF ASC(A$)>44 AND ASC(A$)<58 THEN B
$="V"
3280 RETURN
3290 REM ***********************
3300 REM ***DEFINE***
3310 A$=MID$(A$,9)
3320 F$=LEFT$(A$,X(2)-9)
3330 G$=MID$(A$,X(4)-6)
3340 J=0
3350 J=J+1
3360 IF MID$(G$,J,1)=" " THEN 3390
3370 IF J<LEN(G$) THEN 3350
3380 B$=" DEFINE INCORRECTO":RETURN
3390 G$=LEFT$(G$,J-1)
3400 NWDS=NWDS+1
3410 O$(NWDS)=G$:N$(NWDS)=F$
3420 B$=F$
3430 RETURN
3450 REM ****INICIALIZACION****
3460 PRINT "":POKE 53280,0:POKE53281,0
3470 DIM G$(20),O$(20),N$(20),X(12),Y(12
),Z(12)
3480 NWDS=0:REM * CONTADOR DE FUNCIONES
DEFINIDAS POR EL USUARIO *
3490 GOTO 70
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP 3
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200 STEP 3
```

```
4120 POKE SID+1,L1
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -3
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

La versión del programa PROLOG-A para Commodore 64 utiliza las siguientes órdenes:

MSX	*COMMODORE*	*MSX*	*COMMODORE*	*MSX*	*COMMODORE*
METE	ADD	VECES	TIMES	CUAL	WHICH
SI	IF	MENOR	LESS	LISTA	LIST
Y	AND	ENT	INT	LISTA TODO	LIST ALL
SUMA	SUM	ES	IS	Z	Y

También es necesario utilizar "." en lugar de ":", por ejemplo:

WHICH(X : SUM(1 1 2)) será WHICH(X . SUM(1 1 2))

PROLOG-A

```
1 REM ****************
2 REM *              *
3 REM *   PROLOG-A   *
4 REM *              *
5 REM ****************
10 REM
20 REM *TODAS LAS INTRODUCCIONES EN MA-
      YUSCULAS*
30 GOTO 50
40 PRINT"
        NO HAY (MAS) RESPUESTAS":RETURN
50 GOSUB 3270:REM INICIALIZACION
60 REM ******************************
70 GOSUB 4000:PRINT
80 J$="":INPUT "      &.";J$
90 IF J$="" THEN END
100 PRINT:PRINT "";:IF J$="LIST ALL" THE
N GOSUB 1860:GOTO 70
110 IF LEFT$(J$,5)="LIST " THEN J$=MID$(
J$,5)+" ":GOSUB 990:GOTO70
120 IF RIGHT$(J$,1)<>")" THEN PRINT"1.":
INPUT M$:J$=J$+M$:GOTO 120
130 LJ=LEN(J$)
140 J$=LEFT$(J$,LJ-1)+" ":REM SUSTITU-
     CION DEL ULTIMO ')' POR UN ESPACIO
150 LJ=LEN(J$)
160 FLAG=0
170 IF LEFT$(J$,4)="ADD(" THEN J$=MID$(J
$,5):FLAG=1
180 RULEFLAG=0:PLUSFLAG=0:ARITHFLAG=0
190 FOR R=1 TO LEN(J$)
200 IF MID$(J$,R,4)=" IF " THEN RULEFLAG
=R:FLAG=6
210 IF MID$(J$,R,5)=" AND " THEN PLUSFLA
G=R
220 IF MID$(J$,R,4)="SUM(" THEN ARITHFLA
G=1
230 IF MID$(J$,R,6)="TIMES(" THEN ARITHF
LAG=2
240 IF MID$(J$,R,6)=" LESS " THEN ARITHF
LAG=3
250 IF MID$(J$,R,3)="INT" THEN ARITHFLAG
=4
260 NEXT R
270 IF LEFT$(J$,3)="IS(" THEN J$=MID$(J$
,4):FLAG=2
280 IF LEFT$(J$,10)="WHICH(X . " THEN J$
=MID$(J$,11):FLAG=3
290 IF LEFT$(J$,16)="WHICH((X Y) . X " T
HEN J$=MID$(J$,17):FLAG=4
300 IF FLAG=0 THEN PRINT "ERROR DE SINTA
XIS":GOTO 70
310 LJ=LEN(J$)
320 REM BIFURCACION A LAS SUBRUTINAS SUB
      SIGUIENTES
330 IF PLUSFLAG<>0 THEN GOSUB 1950:GOTO
70 :REM CODIFICACION DE REGLA QUE TIENE
'Y'
340 IF RULEFLAG<>0 AND FLAG<>5 THEN GOSU
B 1110:REM CODIFICACION DE REGLA
350 IF ARITHFLAG<>0 THEN GOSUB 2430:GOTO
 70:REM ARITMETICA
360 IF RIGHT$(J$,3)=" X " OR RIGHT$(J$,3
)=" Y " THEN J$=LEFT$(J$,LJ-2)+" "
370 LJ=LEN(J$)
380 IF FLAG=1 THEN GOSUB 440:REM ADD
390 IF FLAG=2 THEN GOSUB 520:REM IS
400 IF FLAG=3 THEN GOSUB 610:REM WHICH
410 IF FLAG=4 THEN GOSUB 830:REM WHICH2
420 GOTO 70
430 REM **************************
440 REM          **ADD**
450 K=0
460 K=K+1
```

```
470 IF Z$(K)="" THEN Z$(K)=J$:RETURN
480 IF K<1000 THEN 460
490 PRINT "MEMORIA LLENA"
500 RETURN
510 REM ***************************
520 REM          **IS**
530 K=0
540 K=K+1
550 IF Z$(K)="" THEN 580
560 IF Z$(K)=J$ THEN PRINT "SI":GOTO590
570 IF K<1000 THEN 540
580 PRINT "NO"
590 RETURN
600 REM ***************************
610 REM          **WHICH**
620 IF LEFT$(J$,1)="X" THEN 710
630 J$=LEFT$(J$,LJ-1)
640 K=0
650 K=K+1
660 IF Z$(K)="" THEN 690
670 IF J$=LEFT$(Z$(K),LEN(J$)) THEN PRIN
T RIGHT$(Z$(K),(LEN(Z$(K))-LEN(J$)))
680 IF K<1000 THEN 650
690 GOSUB 40
700 RETURN
710 REM  * PREGUNTA INICIADA POR 'X' *
720 J$=MID$(J$,3,LEN(J$)-3)
730 LJ=LEN(J$)
740 K=0
750 K=K+1
760 IF Z$(K)="" THEN 800
770 Q$=MID$(Z$(K),LEN(Z$(K))-LJ,LJ)
780 IF Q$=J$ THEN PRINT LEFT$(Z$(K),LEN(
Z$(K))-(LJ+2))
790 IF K<1000 THEN 750
800 GOSUB 40
810 RETURN
820 REM ***************************
830 REM          **WHICH2**
840 J$=LEFT$(J$,LJ-2)
850 LJ=LEN(J$)
860 K=0
870 K=K+1
880 IF Z$(K)="" THEN 960
890 LFLAG=0
900 FOR L=1 TO (LEN(Z$(K))-LJ)
910 IF MID$(Z$(K),L,LJ)=J$ THEN LFLAG=L
920 NEXT L
930 IF LFLAG=0 THEN 950
940 PRINT LEFT$(Z$(K),LFLAG-2);MID$(Z$(K
),(LFLAG+LJ))
950 IF K<1000 THEN 870
960 GOSUB 40
970 RETURN
980 REM ***************************
990 REM          **LIST**
1000 K=0
1010 K=K+1
1020 IF Z$(K)="" THENRETURN
1030 LFLAG=0
1040 FOR L=1 TO (LEN(Z$(K))-LEN(J$))
1050 IF MID$(Z$(K),L,LEN(J$))=J$ THEN LF
LAG=1
1060 NEXT L
1070 IF LFLAG=1 THEN PRINT Z$(K)
1080 IF K<1000 THEN 1010
1090 RETURN
1100 REM ***************************
1110 REM    FORMACION DE REGLAS
1120 R=RULEFLAG
1130 E$=LEFT$(J$,R):F$=MID$(J$,R+4)
1140 IF LEFT$(E$,1)<>"X" THEN PRINT "ERR
OR EN LA REGLA":GOTO 70
1150 REM LA LINEA SIGUIENTE DETECTA EN-
TRADAS TALES COMO'X COME Y SI X ES-UN Y'
1160 IF RIGHT$(F$,2)="Y " THEN 1390
1170 PRINT:PRINT TAB(10);"COMPILACION DE
 LA REGLA":PRINT
1180 FOR T=1 TO 100
1190 R$(T)=""
1200 NEXT T
1210 E$=MID$(E$,3):F$=MID$(F$,3)
1220 K=0:RR=0
1230 K=K+1
1240 IF Z$(K)="" THEN 1300
1250 IF RIGHT$(Z$(K),LEN(F$))<>F$ THEN 1
370
1260 RR=RR+1
1270 R$(RR)=LEFT$(Z$(K),(LEN(Z$(K))-LEN(
F$)))+E$
1280 PRINT "> ";R$(RR)
1290 GOTO 1230
1300 IF RR=0 THEN RETURN
1310 RC=0
1320 RC=RC+1
1330 Z$(K)=R$(RC)
1340 IF K<1000 THEN K=K+1
1350 IF RC<RR THEN 1320
1360 RETURN
1370 IF K<1000 THEN 1230
1380 RETURN
1390 REM * REGLA CON DOS VARIABLES *
1400 FOR T=1 TO 100
1410 R$(T)=""
1420 NEXT T
1430 K=0:RR=0
1440 IF K=1000 THEN RETURN
1450 K=K+1
1460 IF Z$(K)="" THEN 1770
1470 REM DESCOMPOSICION EN TRES PALABRAS
1480 Q$=Z$(K)
1490 J=0
1500 J=J+1
1510 IF MID$(Q$,J,1)=CHR$(32) THEN 1540
1520 IF J<LEN(Q$) THEN 1500
1530 PRINT "   ERROR EN LA COMPILACION D
E LA REGLA":GOTO 70
1540 A$=LEFT$(Q$,J)
1550 Q$=MID$(Q$,J+1)
1560 J=0
1570 J=J+1
1580 IF MID$(Q$,J,1)=CHR$(32) THEN 1610
1590 IF J<LEN(Q$) THEN 1570
1600 PRINT "   ERROR EN LA COMPILACION D
E LA REGLA":GOTO 70
1610 B$=LEFT$(Q$,J)
1620 Q$=MID$(Q$,J+1)
1630 J=0
1640 J=J+1
1650 IF MID$(Q$,J,1)=CHR$(32) THEN 1680
1660 IF J<LEN(Q$) THEN 1640
1670 PRINT "   ERROR EN LA COMPILACION D
E LA REGLA":GOTO 70
1680 PRINT:PRINT TAB(10);"COMPILACION DE
 LA REGLA" :PRINT
1690 C$=LEFT$(Q$,J)
1700 M$=MID$(F$,3,LEN(B$))
1710 IF B$<>M$ THEN 1440
1720 RR=RR+1
1730 N$=MID$(E$,3,LEN(E$)-4)
1740 R$(RR)=A$+N$+C$
1750 PRINT "> ";R$(RR)
1760 GOTO 1440
1770 IF RR=0 THENRETURN
```

```
1780 M=0
1790 M=M+1
1800 IF M>RR THENRETURN
1810 Z$(K)=R$(M)
1820 IF K=1000 THEN PRINT "MEMORIA LLENA
";GOTO 70
1830 K=K+1
1840 GOTO 1790
1850 REM ***************************
1860 REM         * LIST ALL *
1870 PRINT
1880 K=0
1890 K=K+1
1900 IF Z$(K)="" THENRETURN
1910 PRINT Z$(K)
1920 IF K<1000 THEN 1890
1930 RETURN
1940 REM ***************************
1950 REM FORMACION DE REGLAS CON 'Y' DEL
         TIPO SIGUIENTE:
1960 REM (X COME Y SI X ES-UN AVE E Y
         VIENE EN CAJAS)
1970 REM EL ENUNCIADO X DEBE PRECEDER EN
         LA LISTA AL Y PARA TODOS LOS
1975 REM EJEMPLOS A CODIFICAR
1980 REM DESCOMPOSICION EN SECCIONES
1990 J$=MID$(J$,2):REM ALIGERADO DE 'X'
2000 PRINT:PRINT TAB(10);"COMPILACION DE
 LA REGLA" :PRINT
2010 J=1
2020 J=J+1
2030 IF MID$(J$,J,1)=CHR$(32) THEN 2060
2040 IF J<LEN(J$) THEN 2020
2050 PRINT "   ERROR EN LA COMPILACION D
E LA REGLA":RETURN
2060 A$=LEFT$(J$,J):REM RELACION 1
2070 J$=MID$(J$,J+7):REM CENTRADO EN EL
       COMIENZO DE LA SEGUNDA RELACION
2080 J=1:COUNT=0
2090 J=J+1
2100 IF MID$(J$,J,1)=CHR$(32) THEN COUNT
=COUNT+1
2110 IF COUNT=2 THEN 2140
2120 IF J<LEN(J$) THEN 2090
2130 PRINT "   ERROR EN LA COMPILACION D
E LA REGLA":RETURN
2140 B$=LEFT$(J$,J):REM ENUNCIADO 1
2150 C$=MID$(J$,J+6):REM ENUNCIADO 2
2160 IF C$=CHR$(32) THEN PRINT "   ERROR
 EN LA COMPILACION DE LA REGLA":RETURN
2170 REM RASTREO DE LA BASE DE DATOS
2180 FOR T=1 TO 200
2190 R$(T)=""
2200 NEXT T
2210 R1=0:R2=99
2220 K=0
2230 K=K+1
2240 IF Z$(K)="" THEN 2310
2250 IF R1=99 OR R2=200 THEN PRINT "INSU
FICIENCIA DE MEMORIA":GOTO 2310
2260 LB=LEN(B$)
2270 IF RIGHT$(Z$(K),LB)=B$ THEN R1=R1+1
:R$(R1)=LEFT$(Z$(K),LEN(Z$(K))-LB)
2280 LC=LEN(C$)
2290 IF RIGHT$(Z$(K),LC)=C$ THEN R2=R2+1
:R$(R2)=LEFT$(Z$(K),LEN(Z$(K))-LC)
2300 IF K<1000 THEN 2230
2310 IF R$(100)=""THENPRINT"ENUNCIADO2 I
NTRODUCIDO NO CONSTA EN LA  BASE DE DATO
S":RETURN
2320 REM CODIFICACION DE REGLAS
2330 R1=0:R2=99
2340 R2=R2+1
2350 R1=R1+1
2360 IF R$(R1)>CHR$(32) AND R$(R2)>CHR$(
32) THEN Z$(K)=R$(R1)+A$+R$(R2)+" "
2370 PRINT"> ";Z$(K)
2380 K=K+1
2390 IF R$(R2+1)<>"" THEN 2340
2400 IF R$(R1+1)<>"" THEN 2350
2410 RETURN
2420 REM ***************************
2430 REM          ARITHMETIC
2440 LJ=LEN(J$)
2450 IF ARITHFLAG<3 THEN GOSUB 2490
2460 IF ARITHFLAG=3 THEN GOSUB 2890
2470 IF ARITHFLAG=4 THEN GOSUB 3080
2480 RETURN
2490 REM ****** SUM TIMES *********
2500 J$=MID$(J$,5,LJ-5)
2510 IF LEFT$(J$,2)="S(" THEN J$=MID$(J$,3)
2520 LJ=LEN(J$)
2530 K=0
2540 K=K+1
2550 IF MID$(J$,K,1)=CHR$(32) THEN A$=LE
FT$(J$,K-1):J$=MID$(J$,K+1):GOTO 2580
2560 IF K<LJ THEN 2540
2570 PRINT TAB(12);"ERROR ARITMETICO":RE
TURN
2580 LJ=LEN(J$)
2590 K=0
2600 K=K+1
2610 IF MID$(J$,K,1)=CHR$(32) THEN B$=LE
FT$(J$,K-1):J$=MID$(J$,K+1):GOTO 2640
2620 IF K<LJ THEN 2600
2630 PRINT TAB(12);"ERROR ARITMETICO":RE
TURN
2640 LJ=LEN(J$)
2650 K=0
2660 K=K+1
2670 IF MID$(J$,K,1)=CHR$(41) THEN C$=LE
FT$(J$,K-1):GOTO 2700
2680 IF K<LJ THEN 2660
2690 PRINT TAB(12);"ERROR (DEMASIADAS VA
RIABLES)":RETURN
2700 AN=0:BN=0:CN=0
2710 IF ASC(A$)>58 THEN AN=1
2720 IF ASC(B$)>58 THEN BN=2
2730 IF ASC(C$)>58 THEN CN=4
2740 GUIDE=AN+BN+CN:IF GUIDE=3 OR GUIDE=
5 OR GUIDE=6 THEN 2690
2750 IF ARITHFLAG=2 THEN 2820:REM TIMES
2760 IF GUIDE>0 THEN 2790
2770 IF VAL(A$)+VAL(B$)=VAL(C$) THEN PRI
NT "SI":RETURN
2780 PRINT "NO":RETURN
2790 IF GUIDE=1 THEN PRINTVAL(C$)-VAL(B$
):GOSUB 40:RETURN
2800 IF GUIDE=2 THEN PRINTVAL(C$)-VAL(A$
):GOSUB 40:RETURN
2810 PRINT VAL(A$)+VAL(B$):GOSUB 40:RETU
RN
2820 REM **TIMES**
2830 IF GUIDE>0 THEN 2860
2840 IF VAL(A$)*VAL(B$)=VAL(C$) THEN PRI
NT "SI":RETURN
2850 PRINT "NO":RETURN
2860 IF GUIDE=1 THEN PRINTVAL(C$)/VAL(B$
):GOSUB 40:RETURN
2870 IF GUIDE=2 THEN PRINTVAL(C$)/VAL(A$
):GOSUB 40:RETURN
2880 PRINT VAL(A$)*VAL(B$):GOSUB 40:RETU
RN
2890 REM ***********LESS*************
```

```
2900 NF=0
2910 IF ASC(J$)<58 THEN NF=1:REM NUMEROS
2920 COUNT=0
2930 K=0
2940 K=K+1
2950 IF MID$(J$,K,1)=CHR$(32) THEN COUNT
=COUNT+1
2960 IF COUNT=2 THEN 3000
2970 IF K<LEN(J$) THEN 2940
2980 PRINT TAB(12);"ERROR DE COMPARACION"
2990 RETURN
3000 B$=MID$(J$,K+1)
3010 A$=LEFT$(J$,K-6)
3020 IF NF=1 THEN 3050
3030 IF A$<B$ THEN PRINT "SI":RETURN
3040 PRINT "NO":RETURN
3050 REM * NUMEROS *
3060 IF VAL(A$)<VAL(B$) THEN PRINT "SI":
RETURN
3070 PRINT "NO":RETURN
3080 REM ***********INT*************
3090 IF RIGHT$(J$,2)="X " THEN 3190
3100 K=0
3110 K=K+1
3120 IF MID$(J$,K,1)=CHR$(32) THEN 3160
3130 IF K<LEN(J$) THEN 3110
3140 PRINT TAB(12);"ERROR ARITMETICO"
3150 RETURN
3160 A=VAL(LEFT$(J$,K-1))
3170 IF INT(A)=A THEN PRINT "SI":RETURN
3180 PRINT "NO":RETURN
3190 K=0
3200 K=K+1
3210 IF MID$(J$,K,1)=CHR$(32) THEN 3240
3220 IF K<LEN(J$) THEN 3200
3230 PRINT TAB(12);"ERROR ARITMETICO":RE
TURN
3240 PRINT INT(VAL(LEFT$(J$,K-1)))
3250 RETURN
3260 REM ****************************
3270 REM        INICIALIZACION
3280 PRINT "":POKE 53280,0:POKE53281,0
3290 DIMZ$(1000),R$(200)
3300 RETURN
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP 3
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200 STEP 3
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -3
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

FUZZY RITA

```
1 REM *****************
2 REM *               *
3 REM *   FUZZY RITA  *
4 REM *               *
5 REM *****************
10 REM
20 GOSUB 1380:REM INICIALIZACION
30 GOSUB 1160:REM OPCIONES DE SALIDA
40 GOSUB 1250:REM PREGUNTAS DISCRIMINA-
   TORIAS
50 GOSUB 140:REM CONSULTAS AL USUARIO
60 GOSUB 480:REM TOMA DE DECISION Y AC-
   TUALIZACION DEL CONOCIMIENTO BASE
65 GOSUB 2000:REM SONIDO
70 PRINT"           PULSE <RETURN> PARA CON
TINUAR ";
80 IF X$<>"" THEN INPUT I$:GOTO50
90 PRINT"ELADIESTRAMIENTO"
100 PRINT"        O CUALQUIER OTRA TECLA
 Y DESPUES<RETURN> PARA UTILIZAR 'RITA'
110 INPUT X$:GOTO50
120 END
130 REM ****************************
140 REM CONSULTAS AL USUARIO
150 PRINT""
160 PRINT"       LOS SUJETOS QUE PUEDO DI
FERENCIAR SON LOS SIGUIENTES:"
170 PRINT
180 FOR J=1 TO TT
190 PRINT" >    ";A$(J)
200 NEXT J:PRINT "            "
210 GOSUB 1500
220 IF X$="" THEN PRINT "     PIENSE EN U
NO Y A CONTINUACION PULSE <RETURN>"
230 IF X$<>"" THEN PRINT"    ESTOY PREPA
RADO PARA DETERMINAR CUAL HA ELEGIDO"
240 IF X$="" THEN INPUT X$
250 ADD=.5
260 FOR J=1 TO DQ
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X$<>"" AND TT>2 THEN 390:REM COM
PROBACION PARA DETERMINAR SI SE PUEDE
292 REM OMITIR LA PREGUNTA
300 PRINT "        INTRODUZCA UN NUMERO E
NTRE:"
310 PRINT"          1 (VERDAD) Y 0 (FALSO
).             (PARA ACABAR INTRODUZCA
 $)"
320 PRINT:PRINT E$(J);""
330 INPUT H$:IF H$="$" THEN PRINT:PRINT"
GRACIAS":PRINT:END
340 C(J)=VAL(H$)
350 C(J)=ADD*C(J)
360 NEXT J
370 RETURN
380 REM ************************
390 REM COMPROBACION PARA DETERMINAR SI
LA PREGUNTA SE PUEDE OMITIR
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUMP=0
430 NEXTW
440 IF JUMP=0 THEN 300
450 C(JUMP)=B(W,J)
460 GOTO 360
```

```
470 REM **************************
480 REM TOMA DE DECISION
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXTJ
520 ADD=.5
530 FOR J=1 TO TT
540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM PRUEBE CON DISTINTOS COEFICIEN-
        TES EN LAS TRES LINEAS SIGUIENTES
562 REM HASTA OBTENER LOS MEJORES RESUL-
        TADOS
570 IF C(X)=B(J,X) THEN D(J)=D(J)+1
580 IF ABS(C(X)-B(J,X))<.6*ADD THEN E(J)
=E(J)+.4
590 IF ABS(C(X)-B(J,X))<1.2*ADD THEN F(J
)=F(J)+.1
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
690 REM **COMUNICACION DEL RESULTADO**
700 PRINT
710 CFLG=0
720 PRINT "EL RESULTADO MAS PROBABLE ES
";A$(A1)
730 IF A2<>A1 THEN PRINT"EL SIGUIENTE MA
S PROBABLE ES ";A$(A2):CFLAG=1
740 IF A3<>A2 AND A3<>A1 THEN PRINT"EL S
IGUIENTE MAS PROBABLE ES ";A$(A3):CFLAG= 2
750 PRINT
760 PRINT "ES CORRECTO EL RESULTADO MAS
PROBABLE?             (S / N)";
770 INPUT F$:F$=RIGHT$(F$,1)
780 IF F$<>"S" AND F$<>"N" THEN 770
790 IF F$<>"S" AND X$<>"" THEN RETURN
800 IF F$="S" THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 980
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CFLG=0 THEN 890
840 PRINT"ES CORRECTA MI SEGUNDA ELECCIO
N?                     (S / N)   " ;
850 INPUT F$:F$=RIGHT$(F$,1)
860 IF F$="N" THEN 890
870 IF CFLG=1 THEN A1=A2:GOTO 980
880 IF CFLG=2 THEN A1=A3:GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;"- ";A$(J)
920 NEXT J
930 PRINT
940 PRINT"           CUAL ES EL CORRECTO?"
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM     **ADIESTRAMIENTO DE RITA**
    ACTUALIZACION DEL CONOCIMIENTO BASE
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+5*B
(A1,J))/6
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(A1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$="" THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K);B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
1150 REM *************************
1160 REM OPCIONES DE SALIDA
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
1200 PRINT"INTRODUZCA LA OPCION DE SALID
A NUMERO"TT" (PARA ACABAR PULSE <RETURN>
)"
1210 INPUT A$(TT)
1220 IF A$(TT)="" OR TT=51 THEN TT=TT-1:
RETURN
1230 GOTO1180
1240 REM *************************
1250 REM PREGUNTAS DISCRIMINATORIAS
1260 PRINT""
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT"       INTRODUZCA LA PREGUNTA
"DQ:PRINT"     (PARA ACABAR PULSE <RETUR
N>)"
1340 INPUT E$(DQ)
1350 IF E$(DQ)="" OR DQ=51 THEN DQ=DQ-1:
RETURN
1360 GOTO1310
1370 REM *************************
1380 REM INICIALIZACION
1390 PRINT "":POKE 53280,0:POKE 53281,0
1400 REM REDUZCA EN LA LINEA SIGUIENTE E
L TAMANO DE LAS MATRICES CONFORME "
1402 REM A SUS NECESIDADES
1410 DIM A$(50),B(50,50),C(50),D(50),E$(
50),F(50),G(50)
1420 X$=""
1430 PRINT"  SI QUIERE VER EL CONOCIMIEN
TO BASE AC-TUALIZADO ";
1440 PRINT"PULSE CUALQUIER TECLA SEGUIDA
 DE <RETURN>";
1450 PRINT" DESPUES DE CADA EJECUCION."
1460 PRINT"  EN CASO CONTRARIO PULSE SOL
O <RETURN>."
1470 INPUT U$
1480 PRINT ""
1490 RETURN
1500 PRINT"----------------------------
-----------"
1510 RETURN
2000 SID=54272
2010 FOR L1=0 TO 23
2020 POKE SID+L1,0
2030 NEXT L1
2040 POKE SID+24,7
2050 POKE SID+5,15
2060 POKE SID+6,55
2070 POKE SID+4,33
2080 FOR L1=1 TO 40 STEP 4
2090 POKE SID+1,L1
2100 FOR L2=1 TO 10
2110 NEXT L2,L1
2120 POKE SID+6,0
2130 RETURN
```

Programas para Apple II

SSLISP

```
1   REM  *************
2   REM  *           *
3   REM  * S.S. LISP *
4   REM  *           *
5   REM  *************
10   REM
20   REM
30   REM  *************
40   GOTO 4250: REM  ** INICIALIZA
      CION **
50   REM  **********************
60 A$ =  MID$ (A$,E)
70 A$ =  LEFT$ (A$, LEN (A$) - 1)
80   RETURN
90   REM  **********************
100  PRINT
110 NN = 0
120  INPUT " : ";A$
130  IF A$ = "" THEN  END : REM
      ** PARA ACABAR <RTN> **
140  REM  **********************
150  FOR J = 1 TO 12
160 X(J) = 0
170 Y(J) = 0
180 Z(J) = 0
190  NEXT J
200  REM  **********************
210 R = 0:S = 0:T = 0
220 CFIRST = 0:CSECND = 0
230 EDGE = 0
240  FOR J = 1 TO  LEN (A$)
250 B$ =  MID$ (A$,J,1)
260  IF B$ = "(" THEN S = S + 1:Z
     (S) = J: IF T = 0 THEN CFIRS
     T = J
270  IF B$ = ")" THEN T = T + 1:Y
     (T) = J: IF CSECND <  > 0 AND
     EDGE = 0 THEN EDGE = J - 1
280  IF T = 1 AND B$ = ")" THEN C
     SECND = J
290  IF B$ = " " THEN R = R + 1:X
     (R) = J
300  NEXT J
310  IF S = T THEN  GOTO 370: REM
     ** ( ) EQUILIBRADOS **
320  IF S < T THEN  PRINT : PRINT
     "-> OMISION DE ("
330  IF S > T THEN  PRINT : PRINT
     "-> OMISION DE )"
340  INPUT " +  ";B$
350 A$ = A$ + B$
360  GOTO 150
370  IF NWDS = 0 OR NN = 1 THEN  GOTO
     440
380 M$ =  LEFT$ (A$,X(1) - 1)
390  FOR J = 1 TO NWDS
```

```
400  IF M$ = N$(J) THEN A$ = O$(J
     ) +  MID$ (A$, LEN (N$(J)) +
     1)
410  NEXT J
420 NN = 1
430  GOTO 150
440 FLAG = 0
450 B$ = "NADA"
460  IF  LEFT$ (A$,5) = "CAR (" THEN
     FLAG = 1
470  IF  LEFT$ (A$,5) = "CDR (" THEN
     FLAG = 2
480  IF  LEFT$ (A$,6) = "CONS (" THEN
     FLAG = 3
490  IF  LEFT$ (A$,6) = "ATOM (" THEN
     FLAG = 4
500  IF  LEFT$ (A$,7) = "IGUAL ("
      THEN FLAG = 5
510  IF  LEFT$ (A$,6) = "NULO (" THEN
     FLAG = 6
520  IF  LEFT$ (A$,9) = "MIEMBRO
     (" THEN FLAG = 7
530  IF  LEFT$ (A$,8) = "MIGUAL (
     " THEN FLAG = 8
540  IF  LEFT$ (A$,7) = "JUNTA ("
      THEN FLAG = 9
550  IF  LEFT$ (A$,10) = "INVIERT
     E (" THEN FLAG = 10
560  IF  LEFT$ (A$,7) = "MISMO ("
      THEN FLAG = 11
570  IF  LEFT$ (A$,7) = "LISTA ("
      THEN FLAG = 12
580  IF  LEFT$ (A$,8) = "DEFINE (
     " THEN FLAG = 13
590  IF  LEFT$ (A$,6) = "INC1 (" THEN
     FLAG = 14
600  IF  LEFT$ (A$,6) = "DEC1 (" THEN
     FLAG = 15
610  IF  LEFT$ (A$,7) = "CERO? ("
      THEN FLAG = 16
620  IF  LEFT$ (A$,7) = "RESTA ("
      THEN FLAG = 17
630  IF  LEFT$ (A$,5) = "EXP (" THEN
     FLAG = 18
640  IF  LEFT$ (A$,5) = "MAX (" THEN
     FLAG = 19
650  IF  LEFT$ (A$,5) = "MIN (" THEN
     FLAG = 20
660  IF  LEFT$ (A$,6) = "SUMA (" THEN
     FLAG = 21
670  IF  LEFT$ (A$,7) = "MENOS ("
      THEN FLAG = 22
680  IF  LEFT$ (A$,8) = "DIVIDE (
     " THEN FLAG = 23
690  IF  LEFT$ (A$,7) = "RECIP ("
      THEN FLAG = 24
700  IF  LEFT$ (A$,7) = "RESTO ("
      THEN FLAG = 25
710  IF  LEFT$ (A$,6) = "MULT (" THEN
     FLAG = 26
720  IF  LEFT$ (A$,7) = "MAYOR ("
      THEN FLAG = 27
730  IF  LEFT$ (A$,7) = "MENOR ("
      THEN FLAG = 28
740  IF  LEFT$ (A$,11) = "NEGATIV
     O? (" THEN FLAG = 29
750  IF  LEFT$ (A$,9) = "NUMERO?
     (" THEN FLAG = 30
760  IF  LEFT$ (A$,6) = "UNO? (" THEN
     FLAG = 31
770  IF FLAG > 13 THEN  GOTO 800
780  ON FLAG GOSUB 950,1010,1100,
     1250,1350,1490,1550,1690,189
     0,1960,2370,2570,4080
790  GOTO 840
800  IF FLAG > 24 THEN  GOTO 830
810  ON FLAG - 13 GOSUB 2640,2710
     ,2780,2860,3090,3150,3390,34
     30,3490,3560,3620
820  GOTO 840
830  ON FLAG - 24 GOSUB 3710,3770
     ,3830,3890,3950,4020,2780
840  IF FLAG <  > 0 THEN  GOSUB 8
     60
850  GOTO 100
860  REM  *************
870  REM    DEVUELVE
880  REM    RESPUESTA
890  REM  *************
900  PRINT : PRINT "    SU VALOR
     ES..."
910  IF B$ = " C" THEN B$ = " C (
     CIERTO)"
915  IF B$ <  > "()" THEN  PRINT
     B$
920  IF B$ = "()" THEN  PRINT : PRINT
      TAB( 5);"NADA"
930  RETURN
940  REM  *********************
950  REM         **  CAR  **
960  REM  *********************
970  IF S = 2 THEN B$ =  MID$ (A$
     ,Z(2) + 1,X(2) - Z(2))
980  IF S > 2 THEN B$ =  MID$ (A$
     ,CFIRST,CSECND - CFIRST + 1)
990  RETURN
1000  REM  *********************
1010  REM         **  CDR  **
1020  REM  *********************
1030  GOSUB 950
1040 LB =  LEN (B$) + 7
1050 B$ = "(" +  MID$ (A$,LB,EDGE
      - 1)
1060  IF  RIGHT$ (B$,2) = "))" THEN
     B$ =  LEFT$ (B$, LEN (B$) -
     1)
1070  IF  MID$ (B$,2,1) = " " THEN
     B$ = "(" +  MID$ (B$,3)
1080  RETURN
1090  REM  *********************
1100  REM         ** CONS **
1110  REM  *********************
1120 B$ =  MID$ (A$,7, LEN (A$) -
     1)
1130 J = 0
1140  IF  LEFT$ (B$,1) = "(" THEN
     J = 1
1150 J = J + 1
1160  IF  MID$ (B$,J,1) = "(" THEN
      GOTO 1190
1170  IF J <  LEN (B$) THEN  GOTO
     1150
1180 B$ = " > ERROR EN CONS <": RETURN
1190 LB =  LEN (B$) - 1
1200 B$ = "(" +  LEFT$ (B$,J - 1)
      +  MID$ (B$,J + 1)
1210 B$ =  LEFT$ (B$,LB)
1220  IF  RIGHT$ (B$,2) = " )" THEN
```

```
     B$ =  LEFT$ (B$, LEN (B$) -
     2) + ")"
1230  RETURN
1240  REM  ********************
1250  REM         ** ATOM **
1260  REM  ********************
1270 A$ =  MID$ (A$,7, LEN (A$) -
     1)
1280 J = 0
1290 J = J + 1
1300  IF  MID$ (A$,J,1) = " " OR
     MID$ (A$,J,1) = "(" THEN  RETURN
1310  IF J <  LEN (A$) THEN  GOTO
     1290
1320 B$ = " C"
1330  RETURN
1340  REM  ********************
1350  REM       IGUAL
1360  REM  ********************
1370 E = 5: GOSUB 60
1380 J = 0
1390 J = J + 1
1400  IF  MID$ (A$,J,1) = ")" THEN
     RETURN
1410  IF  MID$ (A$,J,1) = " " THEN
     GOTO 1440
1420  IF J <  LEN (A$) THEN  GOTO
     1390
1430  RETURN
1440 C$ =  LEFT$ (A$,J - 1)
1450 A$ =  MID$ (A$,J + 1)
1460  IF C$ = A$ THEN B$ = " C"
1470  RETURN
1480  REM  ********************
1490  REM      NULO
1500  REM  ********************
1510  IF A$ = "NULO ()" THEN B$ =
     "COMANDO ILEGAL (NULO NECESI
     TA ARGUMENTO)"
1520  IF A$ = "NULO (())" THEN B$
     = " C"
1530  RETURN
1540  REM  ********************
1550  REM         MIEMBRO
1560  REM  ********************
1570 C$ =  MID$ (A$,9)
1580 J = 1
1590 J = J + 1
1600  IF  MID$ (C$,J,1) = ")" OR
     MID$ (C$,J,1) = "(" THEN D$
     =  LEFT$ (C$,J): GOTO 1630
1610  IF J <  LEN (C$) THEN  GOTO
     1590
1620  RETURN
1630 J =  LEN (D$)
1640 J = J + 1
1650  IF  MID$ (C$,J, LEN (D$)) =
     D$ THEN C$ =  LEFT$ (C$, LEN
     (C$) - 1): GOTO 1820
1660  IF J <  LEN (C$) THEN  GOTO
     1640
1670  RETURN
1680  REM  ********************
1690  REM         MIGUAL
1700  REM  ********************
1710 C$ =  MID$ (A$,7)
1720 J = 0
1730 J = J + 1
1740  IF  MID$ (C$,J,1) = " " THEN
      GOTO 1770
1750  IF J <  LEN (A$) THEN  GOTO
     1730
1760  RETURN
1770 D$ =  LEFT$ (C$,J)
1780 C$ =  MID$ (C$,J + 2)
1790 C$ =  LEFT$ (C$, LEN (C$) -
     2) + " "
1800 J = 0
1810 J = J + 1
1820  IF  MID$ (C$,J, LEN (D$)) =
     D$ THEN B$ = "(" +  MID$ (C$
     ,J): GOTO 1850
1830  IF J <  LEN (C$) THEN  GOTO
     1810
1840  RETURN
1850 B$ =  LEFT$ (B$, LEN (B$) -
     1) + ")"
1860  IF  RIGHT$ (B$,3) = ")))" THEN
     B$ =  LEFT$ (B$, LEN (B$) -
     1): GOTO 1860
1870  RETURN
1880  REM  ********************
1890  REM            JUNTA
1900  REM  ********************
1910 B$ =  MID$ (A$,9)
1920 B$ =  LEFT$ (B$,Y(1) - 9) +
     " " +  MID$ (B$,Z(3) - 7)
1930 B$ =  LEFT$ (B$, LEN (B$) -
     1)
1940  RETURN
1950  REM  ********************
1960  REM          INVIERTE
1970  REM  ********************
1980 B$ = ""
1990 A$ =  MID$ (A$,11)
2000 A$ =  LEFT$ (A$, LEN (A$) -
     2)
2010 CT = 0
2020 J = 0
2030 J = J + 1
2040  IF J >  LEN (A$) THEN  GOTO
     2240
2050  IF  MID$ (A$,J,1) = " " THEN
     GOTO 2080
2060  IF  MID$ (A$,J,1) = "(" THEN
     GOTO 2120
2070  GOTO 2030
2080 CT = CT + 1
2090 G$(CT) =  LEFT$ (A$,J - 1)
2100 A$ =  MID$ (A$,J + 1)
2110  GOTO 2020
2120 J = J + 1
2130  IF  MID$ (A$,J,2) = "))" THEN
     GOTO 2320
2140  IF  MID$ (A$,J,1) = ")" THEN
     GOTO 2200
2150  IF J =  LEN (A$) THEN  GOTO
     2170
2160  GOTO 2120
2170 CT = CT + 1
2180 G$(CT) = A$ + ")"
2190  GOTO 2260
2200 CT = CT + 1
2210 G$(CT) =  LEFT$ (A$,J)
2220 A$ =  MID$ (A$,J + 1)
2230  GOTO 2020
2240 CT = CT + 1
2250 G$(CT) = A$
```

```
2260  FOR M = CT TO 1 STEP  - 1
2270 B$ = B$ + G$(M)
2280  IF M > 1 THEN B$ = B$ + " "
2290  NEXT M
2300 B$ = "(" + B$ + ")"
2310  RETURN
2320 CT = CT + 1
2330 G$(CT) =  LEFT$ (A$,J + 1)
2340 A$ =  MID$ (A$,J + 2)
2350  GOTO 2020
2360  REM  ********************
2370  REM             IGUAL
2380  REM  ********************
2390 E = 8
2400  GOSUB 60
2410 M =  ASC (A$)
2420  IF M > 47 AND M < 58 THEN  GOTO
     2900
2430 J = 0
2440 J = J + 1
2450  IF  MID$ (A$,J,2) = ") " THEN
     J = J + 1: GOTO 1440
2460  IF  MID$ (A$,J,3) = ")))" THEN
      GOTO 2500
2470  IF  MID$ (A$,J,2) = "))" THEN
      GOTO 2530
2480  IF J <  LEN (A$) THEN  GOTO
     2440
2490  RETURN
2500 C$ =  LEFT$ (A$,J + 2)
2510 A$ =  MID$ (A$,J + 4)
2520  GOTO 1460
2530 C$ =  LEFT$ (A$,J + 1)
2540 A$ =  MID$ (A$,J + 3)
2550  GOTO 1460
2560  REM  ********************
2570  REM             LISTA
2580  REM  ********************
2590 E = 7
2600  GOSUB 60
2610 B$ = "(" + A$ + ")"
2620  RETURN
2630  REM  ********************
2640  REM             INC1
2650  REM  ********************
2660 E = 7
2670  GOSUB 60
2680 B$ =  STR$ ( VAL (A$) + 1)
2690  RETURN
2700  REM  ********************
2710  REM             DEC1
2720  REM  ********************
2730 E = 7
2740  GOSUB 60
2750 B$ =  STR$ ( VAL (A$) - 1)
2760  RETURN
2770  REM  ********************
2780  REM             CERO?
2790  REM  ********************
2800  IF FLAG = 16 THEN E = 8
2810  IF FLAG = 31 THEN E = 7
2820  GOSUB 60
2830  IF A$ = "0" AND FLAG = 16 OR
     A$ = "1" AND FLAG = 31 THEN
     B$ = " C"
2840  RETURN
2850  REM  ********************
2860  REM   ** DOS ARGUMENTOS *
2870  REM  ********************
2880 E = 13
2890  GOSUB 60
2900 J = 0
2910 J = J + 1
2920  IF  MID$ (A$,J,1) = " " THEN
      GOTO 2950
2930  IF J <  LEN (A$) THEN  GOTO
     2910
2940 B$ = "> ERROR - SOLO UN ARGU
     MENTO <": RETURN
2950 P =  VAL ( LEFT$ (A$,J - 1))
2960 Q =  VAL ( MID$ (A$,J + 1))
2970  IF FLAG = 17 THEN B$ =  STR$
     (P - Q): RETURN
2980  IF FLAG = 23 OR FLAG = 25 THEN
     B = P / Q
2990  IF FLAG = 25 THEN B =  INT
     (.5 + Q * (B -  INT (B)) * 1
     000) / 1000
3000  IF FLAG = 18 THEN B = P ^ Q
3010  IF FLAG = 11 AND P = Q THEN
     B$ = " C"
3020  IF FLAG = 27 AND O > Q THEN
     B$ = " C"
3030  IF FLAG = 28 AND P < Q THEN
     B$ = " C"
3040  IF FLAG = 32 THEN B = P - Q

3050  IF FLAG = 11 OR FLAG > 26 THEN
      RETURN
3060 B$ =  STR$ (B)
3070  RETURN
3080  REM  ********************
3090  REM             EXP
3100  REM  ********************
3110 E = 7
3120  GOSUB 60
3130  GOTO 2900
3140  REM  ********************
3150  REM      ** MAX   MIN **
3160  REM         MAS    VECES
3170  REM  ********************
3180 F$ =  LEFT$ (A$,3)
3190 A$ =  MID$ (A$,6)
3200 CT = 0
3210 FLAG = 0
3220  IF F$ = "MULT" THEN CT = 1
3230 J = 0
3240 J = J + 1
3250  IF  MID$ (A$,J,1) = " " THEN
      GOTO 3280
3260  IF J <  LEN (A$) THEN  GOTO
     3240
3270  IF J =  LEN (A$) THEN FLAG =
     1
3280 P =  VAL ( LEFT$ (A$,J - 1))
     : IF FLAG = 0 THEN A$ =  MID$
     (A$,J + 1)
3290  IF FLAG = 0 THEN A$ =  MID$
     (A$,J + 1)
3300  IF F$ <  > "SUMA" AND CT =
     0 THEN CT = P
3310  IF F$ = "MAX" AND P > CT THEN
     CT = P
3320  IF F$ = "MIN" AND P < CT THEN
     CT = P
3330  IF F$ = "SUMA" THEN CT = CT
      + P
3340  IF F$ = "MULT" THEN CT = CT
```

```
     * P
3350  IF FLAG = 0 THEN  GOTO 3230
3360 B$ =  STR$ (CT)
3370  RETURN
3380  REM  *********************
3390  REM          ** MIN **
3400  REM  *********************
3410  GOTO 3180
3420  REM  *********************
3430  REM            SUMA
3440  REM  *********************
3450 F$ = "PLUS"
3460 A$ =  MID$ (A$,7)
3470  GOTO 3200
3480  REM  *********************
3490  REM            MENOS
3500  REM  *********************
3510 E = 8
3520  GOSUB 60
3530 B$ =  STR$ ( -  VAL (A$))
3540  RETURN
3550  REM  *********************
3560  REM          DIVIDE
3570  REM  *********************
3580 E = 11
3590  GOSUB 60
3600  GOTO 2900
3610  REM  *********************
3620 RECIP
3630  REM  *********************
3640 E = 8
3650  GOSUB 60
3660  IF A$ = "0" THEN B$ = "> ER
     ROR - DIVISION POR CERO <": RETURN
3670 B = 1 / ( VAL (A$))
3680 B$ =  STR$ (B)
3690  RETURN
3700  REM  *********************
3710  REM          RESTO
3720  REM  *********************
3730 E = 12
3740  GOSUB 60
3750  GOTO 2900
3760  REM  *********************
3770  REM            MULT
3780  REM  *********************
3790 F$ = "TIMES"
3800 A$ =  MID$ (A$,8)
3810  GOTO 3200
3820  REM  *********************
3830  REM          MAYOR
3840  REM  *********************
3850 E = 11
3860  GOSUB 60
3870  GOTO 2900
3880  REM  *********************
3890  REM          MENOR
3900  REM  *********************
3910 E = 8
3920  GOSUB 60
3930  GOTO 2900
3940  REM  *********************
3950  REM          NEGATIVO?
3960  REM  *********************
3970 E = 9
3980  GOSUB 60
3990  IF  VAL (A$) < 0 THEN B$ =
     "T"
4000  RETURN
4010  REM  *********************
4020  REM          NUMERO
4030  REM  *********************
4040 A$ =  MID$ (A$,10)
4050  IF  ASC (A$) > 44 AND ACS(A
     $) < 58 THEN B$ = "T"
4060  RETURN
4070  REM  *********************
4080  REM          DEFINE
4090  REM  *********************
4100 A$ =  MID$ (A$,9)
4110 F$ =  LEFT$ (A$,X(2) - 9)
4120 G$ =  MID$ (A$,X(4) - 6)
4130 J = 0
4140 J = J + 1
4150  IF  MID$ (G$,J,1) = " " THEN
     GOTO 4180
4160  IF J <  LEN (G$) THEN  GOTO
     4140
4170 B$ = "> ERROR - DEFINE INCOR
     RECTO <": RETURN
4180 G$ =  LEFT$ (G$,J - 1)
4190 NWDS = NWDS + 1
4200 O$(NWDS) = G$
4210 N$(NWDS) = F$
4220 B$ = F$
4230  RETURN
4240  REM  *********************
4250  REM   * INICIALIZACION *
4260  REM  *********************
4270  HOME
4280  DIM G$(20),O$(20),N$(20),X(
     12),Y(12),Z(12)
4290 NWDS = 0: REM

        ** CONTADOR DE FUNCIONES
            DEFINIDAS POR EL
                USUARIO      **

4300  GOTO 100
```

PROLOG-A

```
1   REM   ***********
2   REM   *         *
3   REM   * PROLOG-A *
4   REM   *         *
5   REM   ***********
10  REM
30  GOTO 50
40  PRINT : PRINT "NO HAY (MAS) R
     ESPUESTAS ": RETURN
50  GOSUB 3280: REM  INICIALIZACI
```

```
    ON
70  PRINT
80  INPUT "&. ";J$
90  IF J$ = "" THEN  END
100  IF J$ = "LISTA TODO" THEN  GOSUB
     1860: GOTO 70
110  IF  LEFT$ (J$,6) = "LISTA " THEN
     J$ =  MID$ (J$,6) + " ": GOSUB
     990: GOTO 70
120  IF  RIGHT$ (J$,1) <  > ")" THEN
      PRINT : PRINT "1.";: INPUT
     " ";M$:J$ = J$ + M$: GOTO 12
     0
130 LJ =  LEN (J$)
140 J$ =  LEFT$ (J$,LJ - 1) + " "
     : REM
         *** SUSTITUCION DEL ***
         *** ULTIMO ')' POR  ***
         ***   UN ESPACIO    ***
150 LJ =  LEN (J$)
160 FLAG = 0
170  IF  LEFT$ (J$,5) = "METE(" THEN
     J$ =  MID$ (J$,6):FLAG = 1
180 RULEFLAG = 0:PLUSFLAG = 0:ARI
     THFLAG = 0
190  FOR R = 1 TO  LEN (J$)
200  IF  MID$ (J$,R,4) = " SI " THEN
     RULEFLAG = R:FLAG = 6
210  IF  MID$ (J$,R,3) = " Y " THEN
     PLUSFLAG = R
220  IF  MID$ (J$,R,5) = "SUMA(" THEN
     ARITHFLAG = 1
230  IF  MID$ (J$,R,6) = "VECES("
      THEN ARITHFLAG = 2
240  IF  MID$ (J$,R,7) = " MENOR
     " THEN ARITHFLAG = 3
250  IF  MID$ (J$,R,3) = "ENT" THEN
     ARITHFLAG = 4
260  NEXT R
270  IF  LEFT$ (J$,3) = "ES(" THEN
     J$ =  MID$ (J$,4):FLAG = 2
280  IF  LEFT$ (J$,9) = "CUAL(X ;
      " THEN J$ =  MID$ (J$,10):F
     LAG = 3
290  IF  LEFT$ (J$,15) = "CUAL((X
      Z) ; X " THEN J$ =  MID$ (J
     $,16):FLAG = 4
300  IF FLAG = 0 THEN  PRINT : PRINT
     "> ERROR DE SINTAXIS <": GOTO
     70
310 LJ =  LEN (J$)
320  REM  ***********************
     *                         *
     *     BIFURCACION A LAS   *
     *                         *
     *  SUBRUTINAS PERTINENTES *
     *                         *
     ***************************
330  IF PLUSFLAG <  > 0 THEN  GOSUB
     1950: GOTO 70: REM
      **** CODIFICACION DE ****
      ****   DE REGLA QUE  ****
      **** CONTIENE 'AND'  ****
340  IF RULEFLAG <  > 0 AND FLAG <
      > 5 THEN  GOSUB 1110: REM
       **** CODIFICACION ****
       ****   DE REGLA   ****
350  IF ARITHFLAG <  > 0 THEN  GOSUB
     2440: GOTO 70: REM  ARITMETI
     CA
360  IF  RIGHT$ (J$,3) = " X " OR
      RIGHT$ (J$,3) = " Z " THEN
     J$ =  LEFT$ (J$,LJ - 2) + "
     "
370 LJ =  LEN (J$)
380  IF FLAG = 1 THEN  GOSUB 440:
      REM  METE
390  IF FLAG = 2 THEN  GOSUB 520:
      REM  ES
400  IF FLAG = 3 THEN  GOSUB 610:
      REM  CUAL
410  IF FLAG = 4 THEN  GOSUB 830:
      REM  CUAL2
420  GOTO 70
430  REM  *********************
440  REM    METE
441  REM  *********************
450 K = 0
460 K = K + 1
470  IF Z$(K) = "" THEN Z$(K) = J
     $: RETURN
480  IF K < 1000 THEN  GOTO 460
490  PRINT : PRINT "> MEMORIA LLE
     NA <"
500  RETURN
510  REM  *********************
520  REM      ES
521  REM  *********************
530 K = 0
540 K = K + 1
550  IF Z$(K) = "" THEN 580
560  IF Z$(K) = J$ THEN  PRINT : PRINT
     "SI": GOTO 590
570  IF K < 1000 THEN  GOTO 540
580  PRINT : PRINT "NO"
590  RETURN
600  REM  *********************
610  REM  CUAL
611  REM  *********************
620  IF  LEFT$ (J$,1) = "X" THEN
      GOTO 710
630 J$ =  LEFT$ (J$,LJ - 1)
640 K = 0
650 K = K + 1
660  IF Z$(K) = "" THEN  GOTO 690
670  IF J$ =  LEFT$ (Z$(K), LEN (
     J$)) THEN  PRINT  RIGHT$ (Z$
     (K),( LEN (Z$(K)) -  LEN (J$
     )))
680  IF K < 1000 THEN  GOTO 650
690  GOSUB 40
700  RETURN
710  REM  ***********************
     *                         *
     *PREGUNTA COMIENZA POR 'X' *
     *                         *
     ***************************
720 J$ =  MID$ (J$,3, LEN (J$) -
     3)
730 LJ =  LEN (J$)
740 K = 0
750 K = K + 1
760  IF Z$(K) = "" THEN  GOTO 800
770 Q$ =  MID$ (Z$(K), LEN (Z$(K)
     ) - LJ,LJ)
780  IF Q$ = J$ THEN  PRINT : PRINT
      LEFT$ (Z$(K), LEN (Z$(K)) -
     (LJ + 2))
```

```
790   IF K < 1000 THEN  GOTO 750
800   GOSUB 40
810   RETURN
820   REM  **********************
830   REM      CUAL2
835   REM  **********************
840 J$ =  LEFT$ (J$,LJ - 2)
850 LJ =  LEN (J$)
860 K = 0
870 K = K + 1
880  IF Z$(K) = "" THEN  GOTO 960
890 LFLAG = 0
900  FOR L = 1 TO  LEN (Z$(K)) -
     LJ
910  IF  MID$ (Z$(K),L,LJ) = J$ THEN
     LFLAG = L
920  NEXT L
930  IF LFLAG = 0 THEN  GOTO 950
940  PRINT : PRINT  LEFT$ (Z$(K),
     LFLAG - 2); MID$ (Z$(K),(LFL
     AG + LJ))
950  IF K < 1000 THEN  GOTO 870
960  GOSUB 40
970  RETURN
980  REM  **********************
990  REM     LISTA
991  REM  **********************
1000 K = 0
1010 K = K + 1
1020  IF Z$(K) = "" THEN  RETURN
1030 LFLAG = 0
1040  FOR L = 1 TO  LEN (Z$(K)) -
      LEN (J$)
1050  IF  MID$ (Z$(K),L, LEN (J$)
     ) = J$ THEN LFLAG = 1
1060  NEXT L
1070  IF LFLAG = 1 THEN  PRINT : PRINT
     Z$(K)
1080  IF K < 1000 THEN  GOTO 1010

1090  RETURN
1110  REM  **********************
     *                          *
     *   FORMACION DE REGLAS    *
     *                          *
     ****************************
1120 R = RULEFLAG
1130 E$ =  LEFT$ (J$,R):F$ =  MID$
     (J$,R + 4)
1140  IF  LEFT$ (E$,1) <  > "X" THEN
      PRINT : PRINT "> ERROR EN R
     EGLA <": GOTO 70
1150  REM  **********************
     *                          *
     *LA LINEA SIGUIENTE DETECTA*
     *                          *
     *    ENTRADAS TALES COMO   *
     *                          *
     *  X COME Y  SI X ES UN Z  *
     ****************************
1160  IF  RIGHT$ (F$,2) = "Z " THEN
      GOTO 1390
1170  PRINT : PRINT  TAB( 15);"CO
     MPILACION DE LA REGLA"
1180  FOR T = 1 TO 100
1190 R$(T) = ""
1200  NEXT T
1210 E$ =  MID$ (E$,3):F$ =  MID$
     (F$,3)
1220 K = 0:RR = 0
1230 K = K + 1
1240  IF Z$(K) = "" THEN  GOTO 13
     00
1250  IF  RIGHT$ (Z$(K), LEN (F$)
     ) <  > F$ THEN  GOTO 1370
1260 RR = RR + 1
1270 R$(RR) =  LEFT$ (Z$(K),( LEN
     (Z$(K)) -  LEN (F$))) + E$
1280  PRINT : PRINT "> ";R$(RR)
1290  GOTO 1230
1300  IF RR = 0 THEN  RETURN
1310 RC = 0
1320 RC = RC + 1
1330 Z$(K) = R$(RC)
1340  IF K < 1000 THEN K = K + 1
1350  IF RC < RR THEN  GOTO 1320
1360  RETURN
1370  IF K < 1000 THEN  GOTO 1230

1380  RETURN
1390  REM  **********************
     *                          *
     *   REGLA CON 2 VARIABLES  *
     *                          *
     ****************************
1400  FOR T = 1 TO 100
1410 R$(T) = ""
1420  NEXT T
1430 K = 0:RR = 0
1440  IF K = 1000 THEN  RETURN
1450 K = K + 1
1460  IF Z$(K) = "" THEN  GOTO 17
     70
1470  REM  **********************
     *                          *
     *      DESCOMPOSICION      *
     *                          *
     *     EN TRES PALABRAS     *
     *                          *
     ****************************
1480 Q$ = Z$(K)
1490 J = 0
1500 J = J + 1
1510  IF  MID$ (Q$,J,1) = " " THEN
      GOTO 1540
1520  IF J <  LEN (Q$) THEN  GOTO
     1500
1530  PRINT : PRINT "ERROR EN LA
     COMPILACION DE LA REGLA": GOTO
     70
1540 A$ =  LEFT$ (Q$,J)
1550 Q$ =  MID$ (Q$,J + 1)
1560 J = 0
1570 J = J + 1
1580  IF  MID$ (Q$,J,1) = " " THEN
      GOTO 1610
1590  IF J <  LEN (Q$) THEN  GOTO
     1570
1600  PRINT : PRINT "ERROR EN LA
     COMPILACION DE LA REGLA": GOTO
     70
1610 B$ =  LEFT$ (Q$,J)
1620 Q$ =  MID$ (Q$,J + 1)
1630 J = 0
1640 J = J + 1
1650  IF  MID$ (Q$,J,1) = " " THEN
      GOTO 1680
1660  IF J <  LEN (Q$) THEN  GOTO
     1640
```

```
1670  PRINT : PRINT "ERROR EN LA
     COMPILACION DE LA REGLA": GOTO
     70
1680  PRINT : PRINT  TAB( 15);"CO
     MPILACION DE LA REGLA"
1690 C$ =  LEFT$ (Q$,J)
1700 M$ =  MID$ (F$,3, LEN (B$))
1710  IF B$ <  > M$ THEN  GOTO 14
     40
1720 RR = RR + 1
1730 N$ =  MID$ (E$,3, LEN (E$) -
     4)
1740 R$(RR) = A$ + N$ + C$
1750  PRINT "> ";R$(RR)
1760  GOTO 1440
1770  IF RR = 0 THEN  RETURN
1780 M = 0
1790 M = M + 1
1800  IF M > RR THEN  RETURN
1810 Z$(K) = R$(M)
1820  IF K = 1000 THEN  PRINT : PRINT
     "MEMORIA LLENA": GOTO 70
1830 K = K + 1
1840  GOTO 1790
1860  REM *********************
     *                         *
     *       LISTA TODO        *
     *                         *
     ***************************
1870  PRINT
1880 K = 0
1890 K = K + 1
1900  IF Z$(K) = "" THEN  RETURN
1910  PRINT Z$(K)
1920  IF K < 1000 THEN  GOTO 1890
1930  RETURN
1950  REM *********************
     *                         *
     *  FORMACION DE REGLAS CON *
     *                         *
     * 'Y' DEL TIPO SIGUIENTE : *
     *                         *
     ***************************
1960  REM *********************
     * X COME Y SI X ES UN AVE  *
     *    E Y VIENE EN CAJAS    *
     *                         *
     *EL ENUNCIADO X DEBE PRECE-*
     *DER EN LA LISTA AL Y PARA *
     *TODOS LOS EJEMPLOS A COD. *
     ***************************
1980  REM *********************
     *                         *
     *      DESCOMPOSICION     *
     *                         *
     *       EN SECCIONES      *
     *                         *
     ***************************
1990 J$ =  MID$ (J$,2): REM
        *** ALIGERADO DE 'X' ***
2000  PRINT  TAB( 5);"INTRERPRETA
     NDO LA REGLA"
2010 J = 1
2020 J = J + 1
2030  IF  MID$ (J$,J,1) = " " THEN
      GOTO 2060
2040  IF J <  LEN (J$) THEN  GOTO
     2020
2050  PRINT : PRINT "ERROR EN LA
     COMPILACION DE LA REGLA": RETURN
2060 A$ =  LEFT$ (J$,J): REM
        *** RELACION 1 ***
2070 J$ =  MID$ (J$,J + 7): REM
      *** CENTRADO EN EL ***
      *** COMIENZO DE LA ***
      ***   RELACION 2   ***
2080 J = 1
2081 COUNT = 0
2090 J = J + 1
2100  IF  MID$ (J$,J,1) = " " THEN
     COUNT = COUNT + 1
2110  IF COUNT = 2 THEN  GOTO 214
     0
2120  IF J <  LEN (J$) THEN  GOTO
     2090
2130  PRINT : PRINT "ERROR EN LA
     COMPILACION DE LA REGLA": RETURN
2140 B$ =  LEFT$ (J$,J): REM
        *** ENUNCIADO 1 ***
2150 C$ =  MID$ (J$,J + 4): REM
     *** ENUNCIADO 2 ***
2160  IF C$ = " " THEN  PRINT : PRINT
     "ERROR EN LA COMPILACION DE
     LA REGLA": RETURN
2170  REM *********************
     *                         *
     *   RASTREO DE LA BASE    *
     *                         *
     *        DE DATOS         *
     *                         *
     ***************************
2180  FOR T = 1 TO 200
2190 R$(T) = ""
2200  NEXT T
2210 R1 = 0
2211 R2 = 99
2220 K = 0
2230 K = K + 1
2240  IF Z$(K) = "" THEN  GOTO 23
     10
2250  IF R1 = 99 OR R2 = 200 THEN
      PRINT : PRINT "INSUFICIENCI
     A DE MEMORIA": GOTO 2310
2260 LB =  LEN (B$)
2270  IF  RIGHT$ (Z$(K),LB) = B$ THEN
     R1 = R1 + 1:R$(R1) =  LEFT$
     (Z$(K), LEN (Z$(K)) - LB)
2280 LC =  LEN (C$)
2290  IF  RIGHT$ (Z$(K),LC) = C$ THEN
     R2 = R2 + 1:R$(R2) =  LEFT$
     (Z$(K), LEN (Z$(K)) - LC)
2300  IF K < 1000 THEN  GOTO 2230
2310  IF R$(100) = "" THEN  PRINT
     : PRINT "EL ENUNCIADO 2 INTR
     ODUCIDO NO FIGURA EN": PRINT
     : PRINT "LA BASE DE DATOS.":
      RETURN
2320  REM *********************
     *                         *
     *  CODIFICACION DE REGLA  *
     *                         *
     ***************************
2330 R1 = 0
2331 R2 = 99
2340 R2 = R2 + 1
2350 R1 = R1 + 1
2360  IF R$(R1) > " " AND R$(R2) >
```

```
     " " THEN Z$(K) = R$(R1) + A$
      + R$(R2) + " "
2370  PRINT : PRINT "> ";Z$(K)
2380 K = K + 1
2390  IF R$(R2 + 1) <  > "" THEN
      GOTO 2340
2400  IF R$(R1 + 1) <  > "" THEN
      GOTO 2350
2410  RETURN
2420  REM **********************
     *                          *
     *        ARITMETICA        *
     *                          *
     ****************************
2440 LJ =  LEN (J$)
2450  IF ARITHFLAG < 3 THEN  GOSUB
     2490
2460  IF ARITHFLAG = 3 THEN  GOSUB
     2890
2470  IF ARITHFLAG = 4 THEN  GOSUB
     3080
2480  RETURN
2490  REM  *** SUMA ***
2500 J$ =  MID$ (J$,5,LJ - 5)
2510  IF  LEFT$ (J$,2) = "S(" THEN
     J$ =  MID$ (J$,3)
2520 LJ =  LEN (J$)
2530 K = 0
2540 K = K + 1
2550  IF  MID$ (J$,K,1) = " " THEN
     A$ =  LEFT$ (J$,K - 1):J$ =
      MID$ (J$,K + 1): GOTO 2580
2560  IF K < LJ THEN 2540
2570  PRINT : PRINT  TAB( 12);"ER
     ROR ARITMETICO": RETURN
2580 LJ =  LEN (J$)
2590 K = 0
2600 K = K + 1
2610  IF  MID$ (J$,K,1) = " " THEN
     B$ =  LEFT$ (J$,K - 1):J$ =
      MID$ (J$,K + 1): GOTO 2640
2620  IF K < LJ THEN  GOTO 2600
2630  PRINT : PRINT  TAB( 12);"ER
     ROR ARITMETICO": RETURN
2640 LJ =  LEN (J$)
2650 K = 0
2660 K = K + 1
2670  IF  MID$ (J$,K,1) = ")" THEN
     C$ =  LEFT$ (J$,K - 1): GOTO
     2700
2680  IF K < LJ THEN  GOTO 2660
2690  PRINT : PRINT  TAB( 12);"ER
     ROR (DEMASIADAS VARIABLES)":
      RETURN
2700 AN = 0:BN = 0:CN = 0
2710  IF  ASC (A$) > 58 THEN AN =
     1
2720  IF  ASC (B$) > 58 THEN BN =
     2
2730  IF  ASC (C$) > 58 THEN CN =
     4
2740 GUIA = AN + BN + CN: IF GUIA
      = 3 OR GUIA = 5 OR GUIA = 6
      THEN  GOTO 2690
2750  IF ARITHFLAG = 2 THEN  GOTO
     2820: REM    *** TIMES ***
2760  IF GUIA > 0 THEN  GOTO 2790
2770  IF  VAL (A$) +  VAL (B$) =
      VAL (C$) THEN  PRINT : PRINT
     "SI": RETURN
2780  PRINT : PRINT "NO": RETURN
2790  IF GUIA = 1 THEN  PRINT : PRINT
      VAL (C$) -  VAL (B$): GOSUB
     40: RETURN
2800  IF GUIA = 2 THEN  PRINT : PRINT
      VAL (C$) -  VAL (A$): GOSUB
     40: RETURN
2810  PRINT : PRINT  VAL (A$) +  VAL
     (B$): GOSUB 40: RETURN
2820  REM ************
     *                 *
     *      TIMES      *
     *                 *
     *******************
2830  IF GUIA > 0 THEN  GOTO 2860
2840  IF  VAL (A$) *  VAL (B$) =
      VAL (C$) THEN  PRINT : PRINT
     "SI": RETURN
2850  PRINT : PRINT "NO": RETURN
2860  IF GUIA = 1 THEN  PRINT : PRINT
      VAL (C$) /  VAL (B$): GOSUB
     40: RETURN
2870  IF GUIA = 2 THEN  PRINT : PRINT
      VAL (C$) /  VAL (A$): GOSUB
     40: RETURN
2880  PRINT : PRINT  VAL (A$) *  VAL
     (B$): GOSUB 40: RETURN
2890  REM ************
     *                 *
     *      MENOS      *
     *                 *
     *******************
2900 NF = 0
2910  IF  ASC (J$) < 58 THEN NF =
     1: REM  *** NUMEROS ***
2920 COUNT = 0
2930 K = 0
2940 K = K + 1
2950  IF  MID$ (J$,K,1) = " " THEN
     COUNT = COUNT + 1
2960  IF COUNT = 2 THEN  GOTO 300
     0
2970  IF K <  LEN (J$) THEN  GOTO
     2940
2980  PRINT : PRINT  TAB( 20);"ER
     ROR DE COMPARACION"
2990  RETURN
3000 B$ =  MID$ (J$,K + 1)
3010 A$ =  LEFT$ (J$,K - 6)
3020  IF NF = 1 THEN  GOTO 3050
3030  IF A$ < B$ THEN  PRINT : PRINT
     "SI": RETURN
3040  PRINT : PRINT "NO": RETURN
3050  REM  *** NUMEROS ***
3060  IF  VAL (A$) <  VAL (B$) THEN
      PRINT : PRINT "SI": RETURN
3070  PRINT : PRINT "NO": RETURN
3080  REM ************
     *                 *
     *       INT       *
     *                 *
     *******************
3090  IF  RIGHT$ (J$,2) = "X " THEN
      GOTO 3190
3100 K = 0
3110 K = K + 1
3120  IF  MID$ (J$,K,1) = " " THEN
      GOTO 3160
```

```
3130  IF K <  LEN (J$) THEN  GOTO
     3110
3140  PRINT : PRINT  TAB( 20);"ER
     ROR ARITMETICO"
3150  RETURN
3160 A1 =  VAL ( LEFT$ (J$,K - 1)
     )
3170  IF  INT (A1) = A1 THEN  PRINT
     : PRINT "SI": RETURN
3180  PRINT : PRINT "NO": RETURN
3190 K = 0
3200 K = K + 1
3210  IF  MID$ (J$,K,1) = " " THEN
     GOTO 3240
3220  IF K <  LEN (J$) THEN  GOTO
     3200
3230  PRINT : PRINT  TAB( 20);"ER
     ROR ARITMETICO": RETURN
3240  PRINT : PRINT  INT ( VAL ( LEFT$
     (J$,K - 1)))
3250  RETURN
3260  REM ***********************
     *                           *
     *       INICIALIZACION      *
     *                           *
     *****************************

3280  HOME
3290  DIM Z$(1000),R$(200)
3300  RETURN
```

FUZZY RITA

```
1   REM  **************
2   REM  *            *
3   REM  * FUZZY RITA *
4   REM  *            *
5   REM  **************
10   REM
20   GOSUB 1420: REM  INICIALIZACI
     ON
30   GOSUB 1190: REM  OPCIONES DE
     SALIDA
40   GOSUB 1280 REM  PREGUNTAS DIS
     CRIMINATORIAS
50   GOSUB 140: REM  PREGUNTAS AL
     USUARIO
60   GOSUB 500: REM  TOMA DE DECIS
     ION Y ACTUALIZACION DE CONOC
     IMIENTOS
70   PRINT "PULSA <RETURN> PARA CO
     NTINUAR"
80   IF X$ <  > "" THEN  INPUT I$:
     GOTO 50
90   PRINT "EL ADIESTRAMIENTO, "
100  PRINT "O CUALQUIER TECLA SEG
     UIDA DE <RETURN>   PARA UTIL
     IZAR A RITA"
110  INPUT X$: GOTO 50
120  END
130  REM  **********************
140  REM  PREGUNTAS AL USUARIO
150  HOME
180  PRINT "ESTOS SON LOS OBJETOS
     QUE PUEDO          DISTINGUI
     R:"
200  FOR J = 1 TO TT
210  PRINT A$(J)
220  NEXT J
230  GOSUB 1560
240  IF X$ = "" THEN  PRINT "PIEN
     SA EN UNO Y LUEGO PULSA <RET
     URN>"
250  IF X$ <  > "" THEN  PRINT "E
     STOY LISTO PARA DEDUCIR CUAL
     TIENES EN MENTE"
260  IF X$ = "" THEN  INPUT J$
270 ADD = .5
280  FOR J = 1 TO DQ
290 ADD = ADD + ADD
300  GOSUB 1560
310  IF X$ <  > "" AND TT > 2 THEN
     410: REM  COMPRUEBA SI SE PU
     EDE OMITIR LA PREGUNTA
320  PRINT "PIENSA EN UNO DE ESTO
     S NUMEROS:"
330  PRINT "1 (CIERTO), 0 (FALSO)
     , $ (TERMINAR)"
340  PRINT : PRINT E$(J);
350  INPUT H$: IF H$ = "$" THEN  PRINT
     : PRINT "HASTA LA VISTA": PRINT
     : PRINT : END
360 C(J) =  VAL (H$)
370 C(J) = ADD * C(J)
380  NEXT J
390  RETURN
400  REM  *********************
410  REM  DECIDE SI SE PUEDE OMIT
     IR LA PREGUNTA
420 SA = 1
430  FOR W = 1 TO TT
440  IF  ABS (B(W,J) - B(1,J)) >
     .7 THEN SA = 0
450  NEXT W
460  IF SA = 0 THEN  GOTO 320
470 C(J) = B(W,J)
480  GOTO 380
490  REM  *********************
500  REM  TOMA DECISION
510  FOR J = 1 TO TT
520 D(J) = 0:E(J) = 0:F(J) = 0
530  NEXT J
540 ADD = .5
550  FOR J = 1 TO TT
560 ADD = ADD + ADD
570  FOR X = 1 TO DQ
580  REM  JUEGA CON LOS VALORES D
     E LAS TRES LINEAS SIGUIENTES
     PARA LOGRAR MAYOR EFICACIA
590  IF C(X) = B(J,X) THEN D(J) =
     D(J) + 1
600  IF  ABS (C(X) - B(J,X)) < .6
     * ADD THEN E(J) = E(J) + .4
```

```
610  IF  ABS (C(X) - B(J,X)) < 1.
     2 * ADD THEN F(J) = F(J) + .
     3
620  NEXT X
630  NEXT J
640 A1 = 1:A2 = 1:A3 = 1
650 F1 = 1:F2 = 1:F3 = 1
660  FOR J = 1 TO TT
670  IF D(J) > F1 THEN F1 = D(J):
     A1 = J
680  IF D(J) > F2 THEN F2 = E(J):
     A2 = J
690  IF D(J) > F3 THEN F3 = F(J):
     A3 = J
700  NEXT J
710  REM  PRESENTA EL RESULTADO
720  PRINT
730 CFLG = 0
740  PRINT "EL RESULTADO MAS PROB
     ABLE ES: ": PRINT A$(A1)
750  IF A2 <  > A1 THEN  PRINT "E
     L SIGUIENTE MAS PROBABLE ES:
     ": PRINT A$(A2):CFLG = 1
760  IF A3 <  > A2 THEN  PRINT "E
     L SIGUIENTE MAS PROBABLE ES:
     ": PRINT A$(A3):CFLG = 2
770 .PRINT
780  PRINT "ES CORRECTO EL RESULT
     ADO MAS PROBABLE?"
790  INPUT F$
800  IF F$ <  > "S" AND F$ <  > "
     N" THEN  GOTO 790
810  IF F$ = "S" THEN PER = 50
815  IF F$ <  > "S" THEN PER = 48
     0
820  IF F$ = "S" AND X$ <  > "" THEN
      RETURN
830  IF F$ = "S" THEN  GOTO 1010
840  IF TT = 2 AND A1 = 1 THEN A1
      = 2: GOTO 1010
850  IF TT = 2 THEN A1 = 1: GOTO
     1010
860  IF CFLG = 0 THEN 920
870  PRINT "ES CORRECTA MI SEGUND
     A ELECCION?"
880  INPUT F$
890  IF F$ = "N" THEN 920
900  IF CFLG = 1 THEN A1 = A2: GOTO
     1010
910  IF CFLG = 2 THEN A1 = A3: GOTO
     1010
920  GOSUB 1560
930  FOR J = 1 TO TT
940  PRINT J;" - ";A$(J)
950  NEXT J
960  PRINT
970  PRINT "CUAL ES LA SOLUCION C
     ORRECTA";
980  INPUT A1
990  IF A1 < 1 OR A1 > TT THEN 98
     0
1000  REM  ACTUALIZACION DE LA BA
     SE DE DATOS
1010  FOR J = 1 TO DQ
1020  IF B(A1,J) <  > 0 THEN B(A1
     ,J) = (C(J) + 5.5 * B(A1,J))
      / 6
1030  IF B(A1,J) = 0 THEN B(A1,J)
      = C(J)
1040 B(A1,J) =  INT (10 * B(A1,J)
     ) / 10
1050  NEXT J
1060  PRINT
1070  IF U$ = "" THEN  RETURN
1080  FOR J = 1 TO TT
1090  PRINT : GOSUB 1560
1100  PRINT A$(J)
1110  PRINT
1120  FOR K = 1 TO DQ
1130  PRINT E$(K);B(J,K)
1140  NEXT K
1150  NEXT J
1160  PRINT
1170  RETURN
1180  REM  *********************
1190  REM  OPCIONES DE SALIDA
1200 TT = 0
1210 TT = TT + 1
1220  GOSUB 1560
1230  PRINT "INTRODUCE EL ELEMENT
     O NUMERO ";TT: PRINT "O PULS
     A <RETURN> PARA TERMINAR"
1240  INPUT A$(TT)
1250  IF A$(TT) = "" OR TT = 50 THEN
     TT = TT - 1: RETURN
1260  GOTO 1210
1270  REM  *********************
1280  REM  PREGUNTAS DISCRIMINATO
     RIAS
1290  HOME
1310  FOR J = 1 TO TT
1320  PRINT A$(J)
1330  NEXT J
1340 DQ = 0
1350 DQ = DQ + 1
1360  GOSUB 1560
1370  PRINT "INTRODUCE LA PREGUNT
     A NUMERO ";DQ: PRINT "O PULS
     A <RETURN> SI NO HAY MAS"
1380  INPUT E$(DQ)
1390  IF E$(DQ) = "" OR DQ = 50 THEN
     DQ = DQ - 1: RETURN
1400  GOTO 1350
1410  REM  *******************
1420  REM  INICIALIZACION
1430  HOME
1440  REM  ADAPTA LAS MATRICES DE
      LA SIGUIENTE LINEA DE ACUER
     DO CON TUS NECESIDADES
1450  DIM A$(50),B(50,50),C(50),D
     (50),E(50),E$(50),F(50)
1460 X$ = ""
1470  PRINT "PULSA UNA TECLA Y DE
     SPUES <RETURN>"
1500  PRINT "SI QUIERES VER LOS C
     ONOCIMIENTOS"
1510  PRINT "ADQUIRIDOS DESPUES D
     E CADA RONDA;"
1520  PRINT "EN CASO CONTRARIO, P
     ULSA SOLO <RETURN>"
1530  PRINT : INPUT U$
1540  HOME
1550  RETURN
1560  PRINT "-------------------
     ------------------"
1570  RETURN
```

ANAYA MULTIMEDIA

Colección «MICROINFORMATICA»

Angell, I. O., y Jones, B. J.: DISEÑO DE GRAFICOS Y VIDEOJUEGOS (incluye cassette).

Barnett, G.: EL LIBRO GIGANTE DE LOS JUEGOS PARA COMMODORE 64/128.

Beechhold, Henry F.: EL LIBRO DEL HARDWARE. No destape su ordenador personal... sin leer antes este libro.

Birmingham Educational Computing Centre: INTRODUCCION A LA TECNOLOGIA DE LA INFORMACION. PREINFORMATICA.

Bishop, Peter: PROGRAMACION AVANZADA EN BASIC.

Brown, Peter: PASCAL A PARTIR DEL BASIC.

Cavalcoli, Aldo: EL ORDENADOR PERSONAL: COMO ELEGIRLO Y UTILIZARLO.

Coccione, L., y Winter, G.: LOS ORDENADORES NO MUERDEN.

Dachslager, H.; Hayashi, M., y Zucker, R.: PROGRAMACION EN BASIC: UN METODO PRACTICO.

Dewhirst, J., y Tennison, R.: TU PRIMER LIBRO DEL ZX SPECTRUM.

D'Opazo, J., y Grupo GOLEM: PROGRAMACION EN LOGO.

Durst, J.: «SPRITES» Y GRAFICOS EN LENGUAJE MAQUINA (ZX SPECTRUM).

Galende Domínguez, F.; Sánchez López, A.; Alfaraz López, M. y Sánchez García, J. A.: COMETAS EN TU MICRO: EL HALLEY. Cálculos de órbitas y parámetros de cometas en BASIC.

Gavin, Maurice: ASTRONOMIA: EL UNIVERSO EN TU ORDENADOR.

Gibbons, John P.: PROGRAMACION AVANZADA DEL COMMODORE 64. Ampliación del BASIC y rutinas gráficas.

Greenwood, Gareth: CODIGOS Y CLAVES SECRETAS. Criptografía en BASIC.

Hammond, R.: EL ORDENADOR Y TUS HIJOS.

Hartnell, Tim: EL LIBRO GIGANTE DE LOS JUEGOS PARA ORDENADOR.

Hartnell, Tim: INTELIGENCIA ARTIFICIAL: CONCEPTOS Y PROGRAMAS.

Hartnell, Tim: EL LIBRO GIGANTE DE LOS JUEGOS PARA ZX SPECTRUM.

Hartnell, Tim, y otros: EL LIBRO GIGANTE DE LOS JUEGOS PARA DRAGON.

Hartnell, Tim: EL SUPERLIBRO DE LOS JUEGOS PARA ORDENADOR.

Heller, R. S., y Martin, C. D.: BITS Y BYTES: INICIACION A LA INFORMATICA.

Hollerbach, Lew: MICROINFORMATICA: CONCEPTOS BASICOS.

Hurley, R.: JUEGOS GRAFICOS DE AVENTURA PARA ZX SPECTRUM.

Johnson, David: DESCUBRE LAS MATEMATICAS CON TU MICRO.

Johnston, J.: MICROS: TAMAÑOS, FORMAS Y SABORES.

Johnston, J.: MICROS: BIPS, PITIDOS Y LUCES.

Johnston, J.: MICROS: MENUS, BUCLES Y RATONES.

Kosniowski, Czes: MATEMATICAS DIVERTIDAS EN BASIC.

Kramer, S.: PROGRAMACION AVANZADA DEL ZX SPECTRUM.

Lacey, Andrew: LIBRO GIGANTE DE LOS JUEGOS PARA MSX.
Núñez, Agustín: PROGRAMACION DEL INTERFACE 1 Y MICRODRIVE.
Otero, M. A.; Pueyo, M. A., y Cajaraville, J. A.: PRIMEROS PASOS EN LOGO. El mundo de la tortuga Fan. Libro del profesor. Libro del alumno.
O'Shea, T., y Self, J.: ENSEÑANZA Y APRENDIZAJE CON ORDENADORES. Inteligencia artificial en educación.
Pentiraro, Egidio: EL ORDENADOR EN EL AULA.
Pritchard, Joe: DESCUBRE TU MSX. Programación y aplicaciones.
Pritchard, Joe: LENGUAJE MAQUINA MSX. Introducción y conceptos avanzados.
Pritchard, Joe: RUTINAS EN LENGUAJE MAQUINA PARA AMSTRAD.
Rosso, Vincenzo de: COMO SE PROGRAMAN LOS ORDENADORES.
Sato, T.; Mapstone, P., y Muriel, I.: MSX: GUIA DEL PROGRAMADOR Y MANUAL DE REFERENCIA.
Servello, Fausto: ¿QUE ES LA TELEMATICA?
Snover, S. L., y Spikell, M. A.: JUEGOS MATEMATICOS DE INGENIO EN BASIC.
Thomasson, Don: PROGRAMACION AVANZADA DEL AMSTRAD. Entradas y salidas de la ROM.
Webb, David: LENGUAJE MAQUINA AVANZADO PARA ZX SPECTRUM.
Zaks, Rodnay: EL LIBRO DEL BASIC.

¿Qué es un sistema experto? ¿Puedo explorar este área de la inteligencia artificial con un microordenador? Si te haces estas preguntas, **SISTEMAS EXPERTOS. INTRODUCCION AL DISEÑO Y APLICACIONES** está escrito para ti.

Un sistema experto es un programa de ordenador que contiene saber humano sobre un tema, mantenido de tal forma que los no expertos pueden acceder a él y utilizarlo. Aunque los sistemas expertos pertenecen a los dominios de máxima sofisticación en informática, si sabes un poco de BASIC y tienes uno cualquiera de los microordenadores para los que están preparados los listados de este libro (MSX, SPECTRUM, AMSTRAD, COMMODORE y APPLE II) podrás aprender mucho acerca de las posibilidades y operativa de los sistemas inteligentes.

A lo largo del libro te familiarizarás con los conceptos de "base de conocimientos", "razonamiento difuso", "mecanismos de inferencia", "crecimiento incremental", y construirás varios sistemas expertos reales con los que experimentar hasta el límite de tu imaginación, ingenio y habilidad:

- —MECANICO DE AUTOMOVILES: diagnostica averías en automóviles.
- —MEDICI: chequea la salud del usuario.
- —FUZZY RITA: sistema experto de propósito general.
- —Emulador de HASTE: lenguaje declarativo simple.
- —Emulador BASIC de PROLOG: versión restringida del PROLOG.
- —Emulador BASIC de LISP: versión restringida del LISP.

¡No esperes más; adéntrate con Tim Hartnell en el mundo de los sistemas expertos!

Contiene listados para MSX, SPECTRUM, AMSTRAD, COMMODORE y APPLE II.






Sistemas Expertos
Tim Hartnell


AMSTRAD
CPC
OCR
MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA
ACME
https://acpc.me/